



**Nuno José R. S.
Magalhães da Costa**

**CONTROLO DE UM ROBOT DE SEIS EIXOS USANDO
TRÊS DRIVES MULTIPLEXADOS**



**Nuno José R. S.
Magalhães da Costa**

**CONTROLO DE UM ROBOT DE SEIS EIXOS USANDO
TRÊS DRIVES MULTIPLEXADOS**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Mestrado Integrado de Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Dr. António Pereira de Melo, Professor Catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

O júri

Presidente

Prof. Doutor José Alberto Gouveia Fonseca

Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Vogais

Prof. Doutor Carlos Fernando Couceiro de Sousa Neves

Professor Coordenador do Departamento de Engenharia Mecânica da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria
(Arguente)

Prof. Doutor António Ferreira Pereira de Melo

Professor Catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro
(Orientador)

Agradecimentos

Quero agradecer ao Professor Dr. António Pereira de Melo pela oportunidade de realizar este projecto.

O meu apreço à empresa Selmatron, ao seu gestor Carlos Mendes Gaudêncio e a todos os funcionários pelo apoio prestado.

Quero agradecer à minha noiva pela paciência, apoio e inspiração ao longo de todo este período e aos meus pais pela oportunidade de frequentar a Universidade de Aveiro.

Palavras-chave

Automação industrial, robótica, robô, labview, multiplexagem, drive

Resumo

Actualmente verifica-se uma evolução tecnológica muito rápida no mundo empresarial industrial. Tentar acompanhar esta evolução constitui uma das prioridades para as empresas que, tendo em vista produzir cada vez mais e de um modo mais eficiente possível, têm que aproveitar ao máximo os recursos que já possuem instalados.

No mundo da automação, mais em concreto no dos manipuladores industriais, acompanhar essa evolução é muito dispendioso, principalmente quando se trata de adquirir novos manipuladores e todo o material necessário para os controlar. As considerações anteriores aplicadas à utilização de equipamentos manipuladores, já existentes num parque industrial, possuindo a componente mecânica em boas condições tecnológicas mas a de controlo obsoleta ou possuindo avarias cuja recuperação é excessivamente dispendiosa, originaram a que, perante um caso concreto, se levantassem as seguintes questões: Será possível realizar a multiplexagem de um servo drive (controlador), normalmente usado para controlar apenas o motor de um eixo, de modo a controlar mais que um motor, isto é utilizar o mesmo servo drive em mais de que um eixo? Deverão ser utilizados sinais analógicos ou digitais? Quais serão as vantagens? Será que manter a versão original, que utiliza sinais analógicos, irá reduzir o custo de recuperação/instalação de um manipulador industrial?

O presente trabalho, que visa contribuir para a resposta a algumas destas questões, foi realizado através de uma parceria entre o Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e a empresa Selmatron, proprietária do robot, sediada em Aveiro.

Foi também desenvolvida uma pequena interface, programada em LabVIEW, para se poder testar o movimento do manipulador industrial usando multiplexagem dos servo drives.

keywords

Industrial automation,robotics,LabVIEW, multiplexing, servo drives

abstract

Presently, within industry, we verify a very fast technological evolution and trying to follow this development is a priority for any company wanting to increase the production, including taking advantage of the resources they hold on the shop floor in order to achieve maximum profitability.

In the world of automation, more specifically of industrial robots, following this development is very expensive when it implies the acquisition of new industrial robots and all the accessories necessary to control them.

Having this in mind, some questions can be raised:

Is it possible to realize multiplexing of a servo drive (controller), normally used to control only one axis, to control more than one motor?

Keeping the use of analogue signals will be recommended? What are the advantages? Will it reduce the cost of recovering an industrial manipulator?

The present work aims to contribute to answer some of the above questions. It was accomplished through a partnership between the Department of Electronics, Telecommunications and Informatics at the University of Aveiro and Selmatron, a company based in Aveiro.

A small interface was also developed, programmed in LabVIEW, to test the movement of the industrial robot using multiplexed servo drives.

Índice de conteúdos

1. Introdução	1
1.1 Enquadramento.....	1
1.2 Objectivo	2
1.3 Organização do documento	2
2. Estado da arte	3
3. Manipulador industrial.....	6
4. Unidades de Drive	8
5. Conceitos teóricos.....	10
5.1 Motor	10
5.2 Resolver.....	15
6. Desenvolvimento do projecto.....	19
6.1 Situação inicial.....	19
6.2 Abordagem ao problema	21
6.2.1 Teste inicial	24
6.3 Compact-Rio	27
6.4 Programação	29
6.4.1 Programação em LabVIEW	29
6.4.2 Controlo de um motor.....	33
6.4.3 Controlo de vários motores com o mesmo drive	47
6.5 Modos de funcionamento.....	56
7. Conclusão e trabalho futuro.....	62
Bibliografia	64

Índice de Figuras

Figura 1 – Manipulador industrial	6
Figura 2 – Movimento de um manipulador	7
Figura 3 – Estrutura de um drive	8
Figura 4 – Família de motores AC	11
Figura 5 – Exemplo de um sinal PWM	13
Figura 6 – Descrição da estrutura interna do motor	13
Figura 7 – Motor Trifásico	14
Figura 8 – Esquema de ligação dos sinais de potência e controlo	15
Figura 9 – Funcionamento de um resolver	16
Figura 10 – Sinal seno e co-seno de um resolver	17
Figura 11 – Desfasamento dos sinais seno e co-seno	18
Figura 12 – Alimentação do drive	21
Figura 13 – Activação do drive	22
Figura 14 – Esquema de ligações no drive	23
Figura 15 – Diagrama de ligações da solução apresentada	26
Figura 16 – Compact-Rio	27
Figura 17 – Chassis reconfigurável	28
Figura 18 – Esquema de funcionamento da FPGA	28
Figura 19 – Exemplo de uma carta de comunicação	29
Figura 20 – Exemplo de um front pannel ou painel frontal	30
Figura 21 – Aparência do block diagram ou diagrama de blocos	31
Figura 22 – Ícone de um vi quando usado em outro vi	32
Figura 23 – Programação da FPGA	34
Figura 24 – Estrutura de eventos dinâmicos	34
Figura 25 – Detector de máximo e armazenamento de valores em arrays	36
Figura 26 – Comunicação com a FPGA	38
Figura 27 – Cálculo da posição angular	39
Figura 28 – Detecção de voltas efectuadas	40
Figura 29 – Controlo da velocidade do motor	41
Figura 30 – Função global	42

Figura 31 – Desvio obtido no cálculo do ângulo	44
Figura 32 – Deslocação angular	46
Figura 33 – Diagrama de ligações	47
Figura 34 – Controlo manual	50
Figura 35 – Estrutura temporal exemplificando a primeira tarefa.....	51
Figura 36 – Segunda tarefa da estrutura temporal.....	52
Figura 37 – Terceira tarefa da estrutura temporal.....	53
Figura 38 – Quarta tarefa de estrutura temporal	54
Figura 39 – Interface com o utilizador em modo manual	55
Figura 40 – Interface com o utilizador em modo “segue-me”	56
Figura 41 – Terceira tarefa no modo “segue-me” de funcionamento.....	57
Figura 42 – Segunda tarefa no modo automático de funcionamento.....	58
Figura 43 – Segunda tarefa após a primeira deslocação.....	59
Figura 44 – Cálculo da posição angular em modo automático.....	60
Figura 45 – Envio da posição angular após a primeira deslocação.....	60
Figura 46 – Quarta fase no modo automático	61

Acrónimos

ABB - Asea Brown Boveri

CAN – Controller area network

DC – Direct current

USB – Universal serial bus

AC - Alternating current

PWM - Pulse width modulation

STO - Safely switched off torque

FPGA - Field-programmable gate array

PCI - Peripheral component interconnect

VI - Virtual Instruments

IO - Input output

1. Introdução

1.1 Enquadramento

Nos dias de hoje a competitividade entre empresas é cada vez maior o que leva a que a luta por uma margem de rentabilidade seja uma questão de sobrevivência.

Numa empresa, se uma linha de montagem parar, este facto pode significar a mudança temporária de viabilidade para prejuízo, tornando-se imperativo agir rapidamente na recuperação e manutenção do material avariado ou parcialmente obsoleto.

Numa altura em que o fundo de maneo de uma empresa é cada vez mais escasso e estando dependentes da entrega bem sucedida de encomendas, a aquisição de novo material pode não ser possível, sendo necessário encontrar alternativas a essa despesa.

No mundo dos manipuladores industriais, a aquisição de novo material é extremamente dispendiosa sendo uma das áreas onde melhor se compreende a necessidade de encontrar alternativas que garantam a operacionalidade sem aumentar excessivamente os investimentos, tanto mais que o crédito bancário é excessivamente caro ou inexistente.

Considerando uma situação específica que surgiu numa empresa, foi-nos pedido que efectuássemos este estudo. Tendo em conta o equipamento e a situação inicial em que o robot se encontrava, tentou-se arranjar uma solução para recuperar o funcionamento deste manipulador industrial.

Tal como foi dito anteriormente, este trabalho foi desenvolvido através de uma parceria entre o departamento de electrónica, telecomunicações e informática da Universidade de Aveiro e uma empresa sediada na zona industrial de Aveiro, a Selmatron. Esta empresa, apesar de poucos anos de existência, tem já alguma experiência e conhecimento na área de automação industrial onde foca a sua estratégia de negócio. Todo o trabalho foi desenvolvido nas instalações da empresa, estando nas mesmas. Todo o material usado incluindo o manipulador industrial.

1.2 Objectivos

Com a realização desta dissertação pretende-se criar uma solução para a recuperação de um manipulador industrial não operacional, devido a ter os drives queimados, usando o equipamento mecânico ainda funcional. Também será desenvolvida uma pequena aplicação programada em LabVIEW. Esta aplicação ajudará no controlo e teste de movimentos do manipulador industrial. Por fim, esta solução será estudada do ponto de vista económico, assim como as suas vantagens e desvantagens que se irão revelar importantes durante o projecto.

1.3 Organização do documento

Esta dissertação está estruturada em 7 capítulos, de acordo com a seguinte disposição:

No capítulo dois é feita uma apresentação do estado de arte, expondo a versatilidade dos manipuladores industriais e as suas capacidades, assim como a importância que estes têm para a indústria neste momento.

Também é exposto um estudo económico sobre a recuperação, mostrando se existe viabilidade numa recuperação de um manipulador industrial, quais as suas vantagens e desvantagens.

No capítulo três é exposto o manipulador industrial utilizado nesta dissertação.

No capítulo quatro está descrita a constituição do drive utilizado nesta dissertação.

No capítulo cinco são expostos os fundamentos teóricos por detrás da utilização de motores eléctricos e do dispositivo designado por resolver.

No capítulo seis é descrito todo o procedimento na recuperação do manipulador industrial, toda a sua programação e ligações eléctricas efectuadas.

No capítulo sete

Por fim, no sétimo capítulo são apresentadas as conclusões da dissertação, bem como o trabalho a ser desenvolvido no futuro.

2. Estado de arte

Actualmente, o número de aplicações em que os manipuladores industriais são utilizados, é muito elevado e variado. Estes manipuladores são muito úteis nas empresas devido à sua grande versatilidade. São unidades capazes de realizar um trabalho duro, sem nunca parar nem perder produtividade. Estão aptos a elevar cargas pesadas, existindo modelos que manipulam na ordem do milhar de quilograma. São igualmente capazes de efectuar um trabalho com uma rapidez, exactidão e precisão sobre-humana, da ordem das centésimas de milímetro.

Sendo um dispositivo de elevada importância para uma empresa, garantir o seu correcto funcionamento é de extrema importância. Hoje em dia, existem algumas empresas líderes no mundo dos manipuladores industriais, como por exemplo ABB (Asea Brown Boveri), Kuka, Beckoff. Estas empresas especializadas garantem a manutenção e correcto funcionamento dos seus robots, estando na linha da frente dos avanços tecnológicos nesta área. Mas tudo isto tem um preço, o qual é bastante elevado.

Num mundo industrial onde a competitividade é feroz e os recursos têm, obrigatoriamente, de ser utilizados ao máximo, muitas vezes não é possível para uma empresa adquirir novos acessórios para a actualização ou recuperação de equipamento pré existente. É frequente, num manipulador industrial, a parte mecânica ser robusta e ter uma longevidade de dezenas de anos. É contudo a parte electrónica que, muitas vezes, deixa de funcionar, devido ao envelhecimento e desactualização dos componentes, ficando assim um manipulador industrial, mecanicamente ainda em bom estado de conservação, parado devido a falha nos drivers electrónicos.

Desta forma, aceita-se a necessidade de desenvolver técnicas de recuperação de manipuladores industriais, aproveitando a parte mecânica em boas condições, fazendo-se alterações na parte electrónica. Neste sentido, o custo de aquisição de um novo manipulador reduz-se a apenas à renovação de alguns dispositivos electrónicos.

O manipulador utilizado nesta dissertação constitui um caso semelhante. Sendo um manipulador já bastante antigo (do tipo puma com mais de 10 anos de fabrico) possui uma parte mecânica em boas condições, estando todos os eixos e caixas de desmultiplicação a funcionar correctamente, assim como os motores. A parte de controlo encontrava-se inutilizável. Os drives originais do manipulador eram muito antigos e encontravam-se queimados devido a uma sobrecarga eléctrica. Deste modo, numa tentativa de recuperação, a empresa adquiriu três drives novos, bastante evoluídos na altura da aquisição (2009). Desta maneira, o objectivo desta dissertação consiste na recuperação do manipulador industrial, utilizando o material já existente na empresa Selmatron.

Tentando evitar a aquisição de novos drives e sendo os existentes em número insuficiente (eram necessários seis e existiam três), pensou-se em implementar uma solução envolvendo a multiplexagem dos drives, ou seja, um drive controlar mais do que um motor. Tendo em conta as características dos motores (três motores iguais, dois motores quase iguais e um motor diferente dos outros) e dos drives disponíveis (um drive de 8 amperes, outro de 16 amperes e um terceiro de 20 amperes), foi necessário dividir o manipulador em três partes. A primeira parte será constituída pelos motores responsáveis pelo movimento da garra, pulso e antebraço (motores 2,2 amperes) e driver de 8 amperes. A segunda parte será constituída pelos motores responsáveis pelo movimento do cotovelo (motor 4,8 amperes) e do ombro (motor de 9,3 amperes) accionados pelo driver de 20 amperes. A terceira parte será constituída pelo motor responsável pelo movimento do tronco (motor de 4,8 amperes) accionado pelo driver de 16 amperes.

Posto isto, foi realizado um estudo económico, analisando a viabilidade da recuperação. A lista de material que se apresentava necessário adquirir consistia num controlador, (que a Selmatron propunha ser uma unidade Compact-Rio da National Instruments), quatro cartas de comunicação, um chassis reconfigurável, seis contactores e setenta relés.

Segundo a empresa National Instruments, um controlador Compact-Rio possui um custo, neste momento, de 1949 euros. O chassis reconfigurável um custo de 1099 euros. As cartas têm um custo: 780 euros para a carta de entradas analógicas, 359 euros para a carta de saídas analógicas e 698 euros para as cartas de entradas e saídas digitais. Estimando um

preço de 5 euros para cada relé, obtém-se um custo de 350 euros. Estimando um preço de 15 euros para cada contactor, obtém-se um custo de 90 euros. O custo total do material a utilizar era assim de 5325 euros, valor que a firma considerou aceitável.

Caso se pretendesse adquirir novos drives, cada drive custaria aproximadamente 2000 euros, perfazendo um custo de 6000 euros apenas na aquisição de mais três drives (ficando a existir um por motor), não contabilizando o preço do controlador e de um quadro eléctrico de controlo.

A opção de adquirir um manipulador novo (com características semelhantes ao existente) conjuntamente com todos os componentes necessários, incluindo os drives e o quadro eléctrico de controlo estima-se, de momento, cerca de 20000 euros.

Deste modo conclui-se que do ponto de vista económico, é atraente recuperar manipuladores industriais utilizando multiplexagem dos drives de controlo.

Encontrando-se o manipulador provisoriamente apenas apoiado numa estrutura frágil de madeira e amarrado com cintas que imobilizam os eixos correspondentes ao tronco e ao ombro, apenas nos foi possível utilizar os movimentos correspondentes ao cotovelo, antebraço, pulso e garra. De contrário, o manipulador poder-se-á desequilibrar e cair, sendo quase impossível levantá-lo de novo devido ao seu elevado peso.

Verificou-se também que os drives existentes tinham sido escolhidos erradamente uma vez que o seu interface de comunicação era do tipo CAN-OPEN e não se dispunha, para a unidade de controlo, de um equipamento possuindo esta interface. Apenas se dispunha de uma carta de comunicação com ligação via protocolo Ethercat. Procurou-se junto da Selmatron conseguir outra carta, mas devido ao seu elevado preço, a empresa informou que não era possível. Ficou, deste modo limitada a utilização das placas drive a apenas via sinais analógicos.

Para o controlo de todo o processo de execução, foi facultado um controlador da National Instruments designado por Compact-Rio. Este controlador vem equipado com quatro cartas de comunicação, duas cartas de comunicação analógica e duas cartas de entradas e saídas digitais. Este controlador será apresentado com mais detalhe numa das secções seguintes (6.3).

3. Manipulador Industrial

O manipulador industrial utilizado nesta dissertação possui seis eixos, todos do tipo rotacional como se pode ver na figura 1.



Figura 1 – Manipulador industrial

Em cada eixo, está acoplado um motor eléctrico do tipo DC Brushless da marca Moog. Estes tipos de motor estão descritos com mais pormenor no capítulo 5 versando os conceitos teóricos.

Junto com o manipulador existiam também três drives de controlo da marca Moog.

Devido ao facto de haver apenas três drives disponíveis, teriam que se adquirir mais três unidades ou alternativamente tentar utilizar um drive para mais que um motor. Visualizando a figura 2, que exemplifica o movimento de cada eixo do manipulador, podemos ponderar numa possível solução.

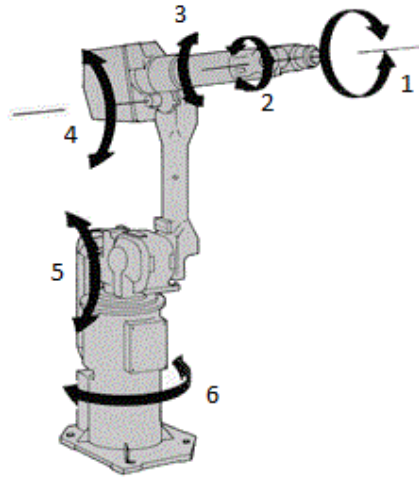


Figura 2 – Movimento de um manipulador

Um drive ficaria encarregue dos três motores responsáveis pelos movimentos 1, 2 e 3. O segundo drive ficaria encarregue do controlo dos motores responsáveis pelos movimentos 4 e 5. O terceiro drive ficaria encarregue do controlo do motor responsável pelo movimento 6.

O manipulador utilizado ao longo desta dissertação consegue manipular objectos com um peso da ordem dos 60 quilograma.

4. Unidades de drive

Os drives são da série G392 da Moog. A característica mais importante que os distingue é a corrente máxima que permite fornecer aos motores. O drive utilizado para motores de menor potência fornece oito amperes na utilização máxima. O drive de potência intermédia, fornece 16 amperes no seu esforço máximo e o drive adequado ao motor de maior potência fornece no máximo 20 amperes.

Estes dispositivos têm um conjunto de funcionalidades que permitem ao programador configurar previamente o modo de funcionamento pretendido, através de uma interface atractiva e simples de se utilizar.

Estruturalmente os três drives são semelhantes, sendo apenas diferentes nas especificações de corrente máxima possível de ser fornecida ao motor.

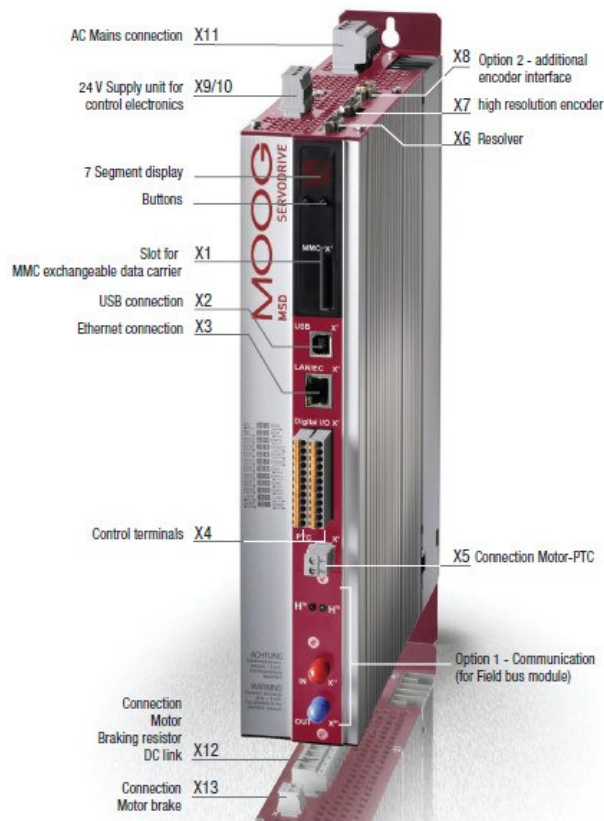


Figura 3 – Estrutura de um drive Moog

Para efeitos de alimentação interna cada drive é alimentado através de uma entrada de 24 volts DC, por onde recebe a alimentação para os sinais de controlo. Existe uma entrada de corrente alternada trifásica onde é entregue a energia que será utilizada para movimentar o veio do motor. Existem ainda três entradas para se ligar um dispositivo de retorno, sendo neste caso um resolver descrito em pormenor no capítulo de conceitos teóricos

Existe ainda um conjunto de entradas e saídas digitais, num total de oito entradas e três saídas, assim como uma carta de comunicação digital via protocolo CAN-OPEN, que não podemos utilizar por o controlador Compact-Rio não dispor de porta de comunicação usando este protocolo.

Para além das entradas digitais, existem também duas entradas analógicas.

Adicionalmente existe a possibilidade de ligação a um computador de diagnóstico através do protocolo Ethernet ou por USB.

Cada drive tem uma saída de 24 volts que permite controlar o travão do motor e possui a saída do sinal de potência que alimentará o motor.

5. Conceitos teóricos

5.1 Motor

Os motores eléctricos transformam energia eléctrica em energia mecânica, fornecendo o binário necessário à parte mecânica.

No mundo dos motores eléctricos existem vários tipos de motores que se dividem em duas grandes áreas, os motores DC e AC.

Os motores AC (alternating current), devido às vantagens da corrente alternada, começaram a ser muito usados a partir do século XIX.

Algumas das suas vantagens são design simples, baixo custo, fiabilidade, podem ser ligados directamente à rede eléctrica.

Estes motores são constituídos basicamente por duas partes: uma parte fixa designada por estator, o qual é alimentado pela corrente alternada para produzir um campo magnético e uma parte móvel designada por rotor ligada ao veio onde será produzido binário.

Dentro da família dos motores AC há vários tipos nomeadamente: motores assíncronos, com escovas, universais, e os síncronos. Dentro do tipo assíncrono subdividem-se em motores com várias fases ou com uma única fase. Dentro do tipo síncrono subdividem-se em motores de onda sinusoidal, motores de passo, motores de relutância e motores “brushless” (sem escovas de carvão).

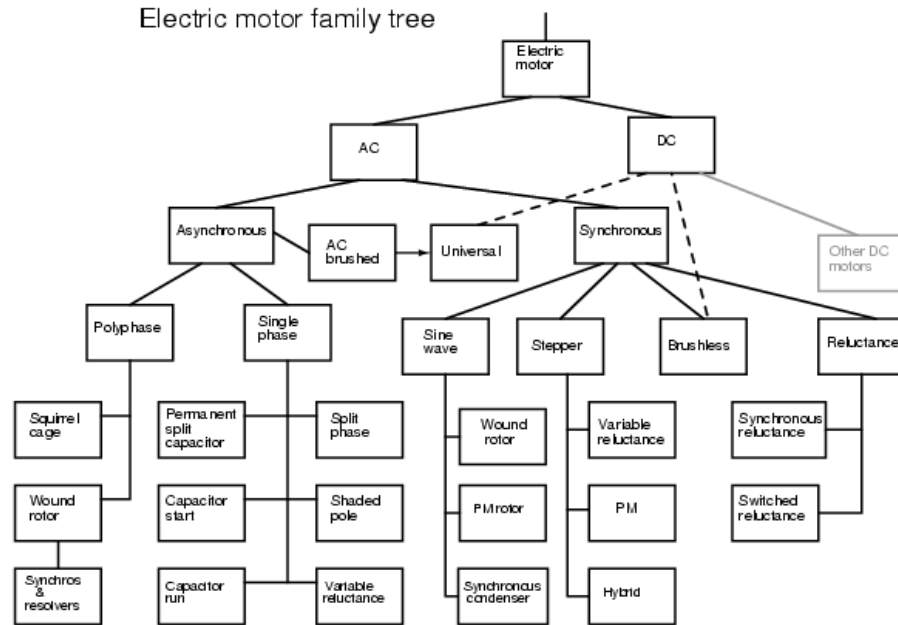


Figura 4 - família dos motores AC

Os motores DC (direct current em inglês) são muito populares hoje em dia, utilizando-se quer nos pequenos dispositivos como brinquedos até às grandes máquinas industriais. São muito usados pois possuem dimensões físicas pequenas e no entanto são capazes de produzir potências elevadas. Podem ser operados a partir de baterias ou ligados à rede eléctrica com as devidas transformações e controlo necessários.

Fisicamente são compostos por duas estruturas: a primeira estrutura designa por campo ou indutor normalmente fixa e outra estrutura designada por armadura ou induzido normalmente móvel. Estes motores podem ser ligados de diferentes formas, as quais se dividem em quatro grandes categorias: ligação em série, ligação em paralelo, ligação composta e ligação independente.

A ligação em série consiste em ligar, como o próprio nome indica, a armadura em série com o campo. Esta ligação tem uma característica interessante em termos de binário do motor. Apresenta um elevado binário de arranque, ou seja o binário é elevado a baixas velocidades angulares e vai diminuindo, não linearmente, à medida que a velocidade vai aumentando. Por outro lado existe a ligação em paralelo onde a ligação entre o campo e a armadura é em paralelo (ou “shunt”). Nesta configuração ocorre o contrário, o binário é

menor a baixas velocidades e aumenta conforme a velocidade também aumenta e é máximo a uma velocidade angular específica.

Na ligação série paralelo temos o melhor dos dois mundos, isto é, temos a ligação em série para fornecer um elevado binário no arranque do motor e depois comuta para a ligação paralela para manter o elevado binário a elevadas velocidades angulares.

A ligação independente é usada quando se deseja um modo de funcionamento linear podendo-se controlar o motor quer pelo indutor quer pelo induzido.

Os motores usados no manipulador industrial são do tipo “brushless”, ou seja, não tem um colectador de lâminas com escovas de carvão, sendo um circuito electrónico que produz a comutação de sentido de corrente necessário no induzido. O indutor é substituído por um íman permanente sendo normalmente a parte móvel, e o induzido a parte fixa.

As vantagens deste tipo de motores são: a pouca ou case nula manutenção, pois não existe contacto físico como acontecia com as escovas de carvão eliminando assim grande parte do desgaste; a alta eficiência e a longa duração de vida. As desvantagens são principalmente no custo inicial na aquisição do motor e a necessidade de um controlador mais complexo que consiga gerir a comutação de corrente.

Vejamos como funciona um motor síncrono “brushless”. Para accionar um motor deste género, normalmente existe um controlador que através de um sinal PWM (pulse width modulation) gera um sinal AC com uma determinada frequência desejada como se pode ver na figura 5. Este sinal AC é gerado da seguinte forma: ligando e desligando o sinal da fonte para a carga a uma velocidade elevada, controla-se a quantidade de corrente e tensão aplicadas na carga (neste caso o motor). Quanto mais tempo estiver o sinal ligado, maior será a tensão e a corrente aplicadas na carga. Quanto menos tempo estiver o sinal ligado, menor será a tensão e a corrente aplicadas na carga. Ajustando o período da onda quadrada PWM, ajusta-se o tempo de aplicação de tensão na carga. Uma das grandes vantagens em usar um sinal PWM é haver perdas praticamente nulas de potência no drive. Quando o sinal está desligado não há praticamente corrente a circular para a carga. Quando está ligado há uma queda quase insignificante da tensão durante o tempo de ligação. Ao contrário dos sistemas lineares em que o nível de tensão é regulado linearmente por uma

resistência variável havendo grandes perdas de potência na dissipação de calor, com o sistema PWM tal não acontece. Durante o tempo de ligação, as únicas resistências presentes são a resistência do interruptor, o qual é responsável por ligar e desligar o sinal, e a resistência parasita presente no filtro, sendo ambas muito reduzidas.

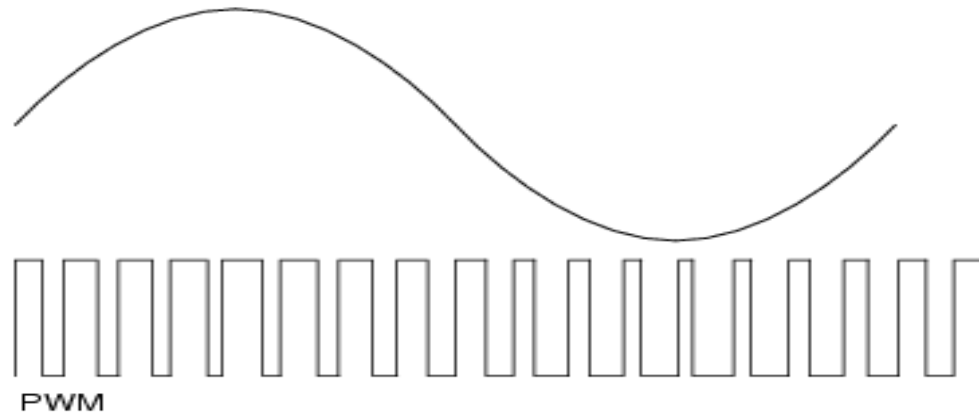


Figura 5 – Exemplo de um sinal PWM

Como se pode ver na figura 6, quando a onda sinusoidal cresce, o pólo norte do induzido repele o pólo norte do indutor e atrai o pólo sul do indutor. Por sua vez, o pólo sul do induzido atrai o pólo norte e repele o pólo sul do indutor, originando assim a rotação do veio do motor. Quando a onda sinusoidal muda de sinal, ou seja, começa a decrescer, ocorre o mesmo fenómeno mas desta vez os pólos no induzido trocam de posição, fazendo assim com que o movimento do eixo seja contínuo.

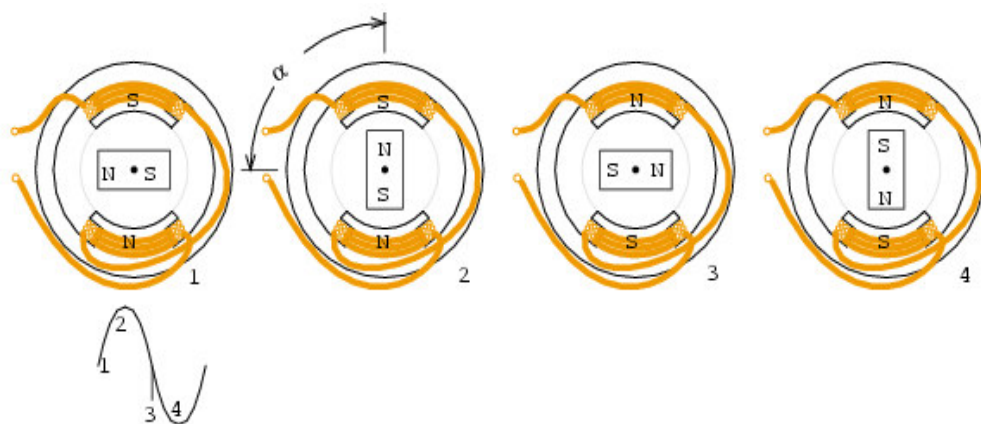


Figura 6 – Descrição da estrutura interna de um motor

No caso do motor do manipulador industrial, não temos apenas uma fase, mas sim três fases. O fenómeno é exactamente o mesmo mas agora com três fases desfasadas de 120°.

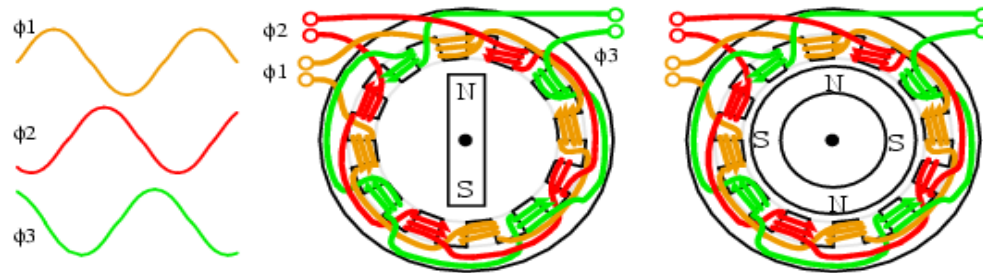


Figura 7 – Motor trifásico

Como podemos ver na figura 7, se ligássemos cada uma das fases sequencialmente, o ímã permanente iria apontar, em sequência, para cada um dos pólos do induzido. Mas, uma vez que as fases se sobrepõem, o movimento do ímã não será brusco mas sim suave na transição de um pólo para o outro, obtendo-se assim um movimento contínuo.

Para se iniciar o movimento é usual o controlador variar a frequência seguindo uma rampa, desde zero até à frequência final. Para o cálculo da frequência necessária, a fim de que o motor rode a uma determinada velocidade, utiliza-se a seguinte expressão:

$$S = (F * 120) * n$$

onde S é a velocidade do motor, F é a frequência e n é o número de pólos do motor.

Para fornecer uma ligação de retorno (feedback), em cada motor está acoplado um resolver. Este dispositivo tem uma resolução bastante elevada por cada revolução do veio do motor. Será visto com mais detalhe no subcapítulo seguinte.

De seguida apresentam-se as características principais dos motores usados.

Motor Moog	Velocidade (rpm)	Binário (Nm)	Binário Max (Nm)	Frequência (Hz)	N.º Pólos	Corrente (A)
D313-043 L05	7200	0,6	1,8	480	8	2,2
D313-045 L25	4200	2,3	6,9	280	8	4,8

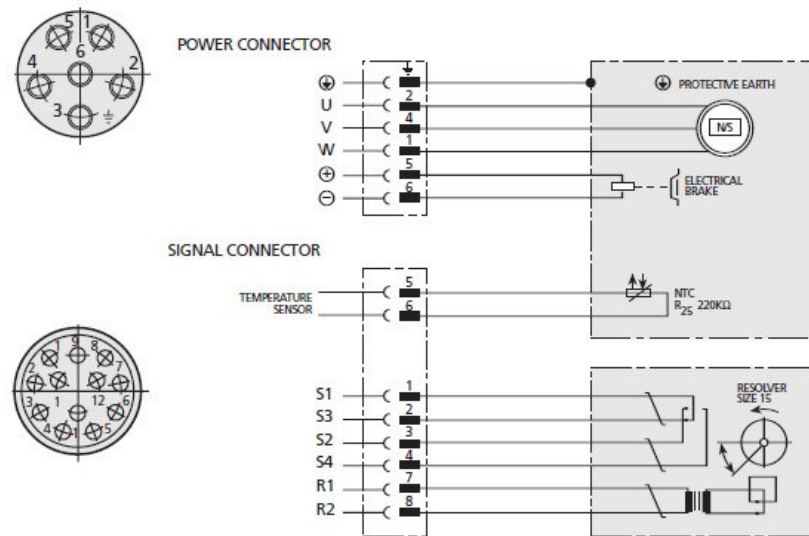


Figura 8 – Esquema de ligação dos sinais de potência e controlo

5.2 Resolver

Como foi referido anteriormente, a cada motor está acoplado um dispositivo que permite fazer a ligação de retorno de velocidade angular do motor ao drive (controlador). Neste caso o dispositivo designa-se por resolver. Este dispositivo tem uma grande resolução detectando pequenas variações angulares. Como princípio de funcionamento temos a figura 8 a demonstrá-lo.

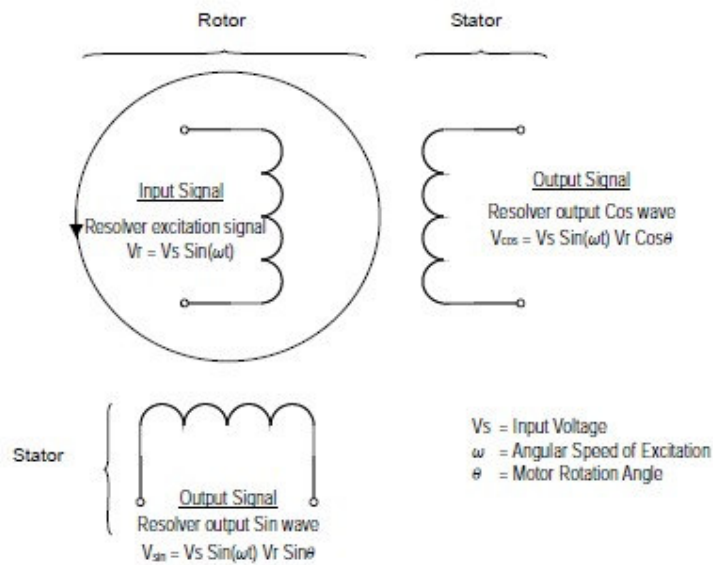


Figura 9 – Funcionamento de um resolver, [10].

Como se pode ver na figura 9 o resolver possui três bobinas. Uma é colocada no rotor, e as restantes são colocadas no estator de modo a estarem posicionadas mecanicamente a 90 graus uma da outra, ou seja ortogonais entre si. Quando o resolver é excitado com um sinal sinusoidal na bobine situada no rotor, dois sinais, seno e co-seno, são induzidos nas bobinas do estator desfasados entre si 90 graus. Esses sinais comportam a frequência da portadora do sinal de excitação e têm amplitude proporcional ao seno e co-seno do ângulo de posição do veio do motor. O sinal de excitação é normalmente um sinal sinusoidal, amplitude constante e frequência fixa na ordem dos kHz (16kHz no presente caso). Quando o veio do motor se desloca, as ondas que se obtêm aos terminais dos enrolamentos do estator são as representadas na figura 10.

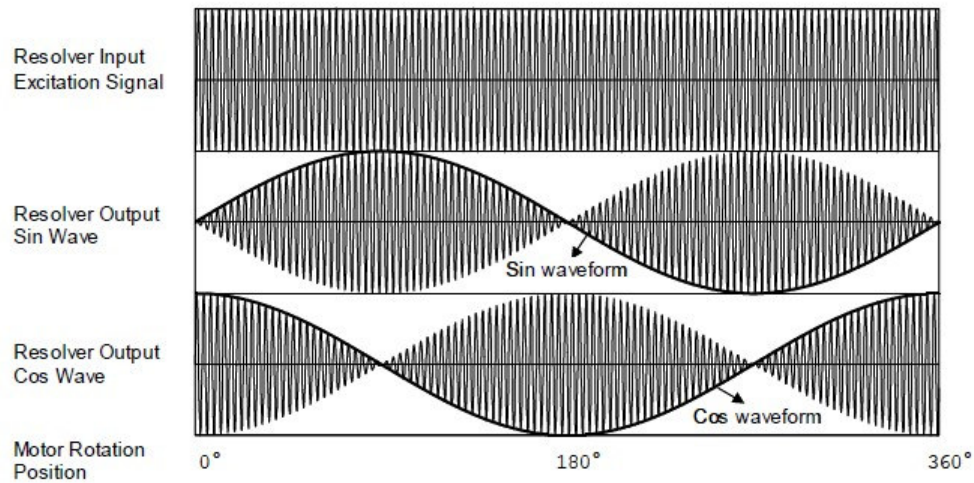


Figura 10 – Sinal seno e co-seno de um resolver, [10].

A cada posição angular do veio do motor, corresponde uma amplitude dos sinais seno e uma amplitude do sinal co-seno que podemos visualizar na figura 10. Note-se que a frequência do sinal de excitação é muito superior à frequência dos sinais envolventes que correspondem ao seno e co-seno de onde se vai obter a posição angular do veio do motor.

O processo de leitura e cálculo do ângulo do veio do motor tendo por base os sinais provenientes do resolver é o que se segue.

Partindo da figura 10, onde se encontram representadas as curvas correspondentes à evolução temporal dos sinais provenientes das saídas do resolver, sinal seno e sinal co-seno (linhas a cheio na figura 10) podemos escrever matematicamente as seguintes expressões:

$$V_{sin} = \alpha A \sin(wt) \sin\theta$$

$$V_{cos} = \alpha A \sin(wt) \cos\theta$$

Onde $A \sin(wt)$ representa o sinal de excitação, α representa o rácio de transformação entre o estator e o rotor e θ representa o ângulo que se pretende determinar. Admitindo que o sinal de excitação tem amplitude e frequência constantes, podemos dividir V_{sin} sobre V_{cos} .

$$\frac{V_{sin}}{V_{cos}} = \frac{\alpha A \sin(\omega t) \sin\theta}{\alpha A \sin(\omega t) \cos\theta}$$

Obtendo assim uma expressão que permite relacionar directamente a posição angular do veio do motor com os sinais Vsin e Vcos,

$$\frac{V_{sin}}{V_{cos}} = \frac{\sin\theta}{\cos\theta}$$

ou seja graficamente o representado na figura 11:

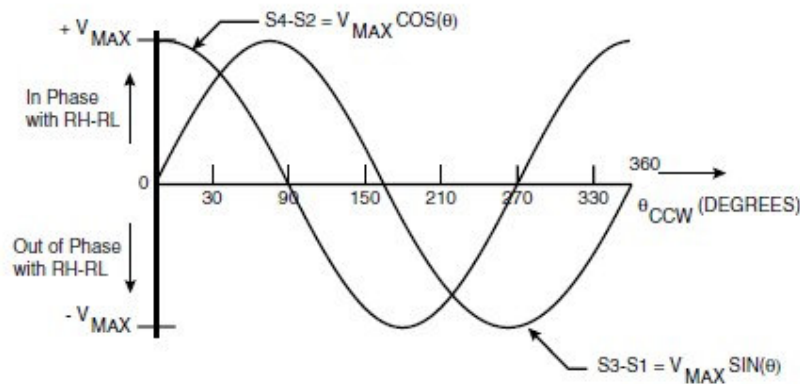


Figura 11 – Desfasamento dos sinais seno e co-seno

de onde se pode tirar:

$$\frac{V_{sin}}{V_{cos}} = \tan\theta$$

ou ainda:

$$\theta = \tan^{-1}\left(\frac{V_{sin}}{V_{cos}}\right)$$

Este método de cálculo assume que para uma determinada posição angular, os sinais Vsin e Vcos possuem a frequência da portadora, isto é, a frequência do sinal de excitação. Não se está a contabilizar possíveis imperfeições, como por exemplo desvios de fase entre os sinais.

6. Desenvolvimento do projecto

6.1 Situação inicial

Inicialmente o material disponível consistia num conjunto de três servo drives Moog, uma unidade controlo programável Compact-Rio da National Instruments, um chassis de um robot de seis eixos do tipo Puma equipado com 6 motores Moog.

Começou-se por averiguar em que estado se encontrava o robot. Ao inspeccionar o robot, verificou-se que seria necessário refazer todas as ligações eléctricas já existentes devido ao mau estado em que se encontravam e à sua desorganização. Identificou-se os componentes que se encontravam no robot, mais em concreto todos os sensores de fim de curso. Verificou-se o estado em que estes se encontravam para posterior encomenda de componentes novos caso fosse necessário. Constatou-se que alguns componentes não funcionavam, tendo sido substituídos posteriormente.

Retiraram-se todos os cabos de ligação existentes, verificou-se quais os tipos de fichas de ligação utilizadas. Em cada motor são usadas duas fichas, uma destinada aos sinais de controlo e outra responsável pelo fornecimento de potência proveniente do servo drive para o motor. Desde logo deparou-se com um problema; qual a função de cada um dos pinos das fichas isto é que sinais conduziam?

O robot é accionado por seis motores bastante antigos cujas características ainda são visíveis na chapa fixa na parte exterior de cada motor. Contactamos a empresa Moog para fornecerem toda a informação que tivessem sobre os motores. Infelizmente, sendo os motores já bastante antigos, a informação disponível era muito reduzida, não contendo a disposição dos pins das fichas de ligação. Após uma longa pesquisa, descobriu-se que um motor Moog mais recente ainda mantinha a mesma disposição dos pins na ficha de ligação.

O passo seguinte foi considerar que tipo de cabos se iria utilizar tanto na ligação de potência como na ligação de sinais de controlo. Para os sinais de controlo teria de ser um cabo com blindagem para minimizar a influência do ruído externo. Então dentro, do

material disponível na empresa, escolheu-se um cabo blindado. Efectuou-se a montagem de todas as fichas de ligação para os sinais de controlo em todos os motores e introduziram-se todos os cabos dentro do chassis do robot.

Para a transmissão de potência para os motores escolheu-se um cabo de 4x4mm de secção, possuindo três condutores para as fases e um condutor de ligação à terra. Este também possui blindagem para impedir que o ruído se propague dentro do chassis do manipulador.

Após todos os cabos serem introduzidos no chassis do robot, procedeu-se à montagem das fichas que ligam todos os sinais de controlo e de potência do robot ao quadro eléctrico de controlo.

Após concluídas as ligações de todas as fichas aplicou-se um sinal de 24 volts para testar o funcionamento do travão de cada motor (apenas nos motores que não suportam a estrutura do robot, isto é, o motor responsável pelo movimento da garra, pulso, antebraço e cotovelo, pois os correspondentes ao ombro e ao tronco não podiam ser movidos por desequilibrarem a estrutura uma vez que esta não estava fixa ao chão) e verificar se o veio rodava em boas condições, pois havia a possibilidade de o sistema de travão estar avariado, ou o veio estar colado no interior, sendo impossível utilizar os motores. Verificou-se que todos os motores testados aparentavam estar em boas condições de funcionamento.

Após estas verificações, passamos a analisar os drives. Constatou-se que apenas existiam três drives em funcionamento, estando os restantes drives inutilizáveis. Também se verificou que nos drives funcionais apenas seria possível usar a parte de comunicação analógica.

Nesta altura colocou-se a questão, seria possível realizar multiplexagem dos drives apenas com sinais analógicos?

6.2 Abordagem ao problema

Como primeiro passo para a abordagem ao problema, decidiu-se por colocar um drive em funcionamento com um motor, estando o motor removido do chassis do robot para se poder testar em segurança. Deparamo-nos então com o problema de como inicializar o drive.

Como referido anteriormente, os drives existentes são da marca Moog. Juntamente com o drive, vem um software para instalação num computador de diagnóstico com o qual é possível supervisionar possíveis erros de funcionamento, alarmes, configuração de alguns parâmetros essenciais para o correcto funcionamento deste.

A ligação do drive ao computador pode ser através de um cabo USB ou uma ligação Ethernet. Para o arranque do drive é necessário alimentá-lo com 24 volts (figura 12). Este sinal irá alimentar toda a parte de controlo do drive, estando a parte de potência separada por questões de segurança.

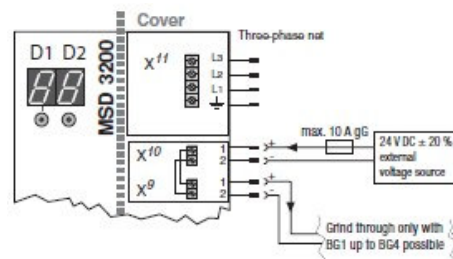


Figura 12 – Alimentação do drive, manual de utilizador Moog.

Após a activação da alimentação do drive, foi necessário proceder à configuração do mesmo de modo a fornecer potência e sinais adequados ao motor a ensaiar. Nesta altura surgiu outra questão, quais as características do motor? Visto que na chapa de identificação apenas faz referência ao torque, velocidade, corrente e inércia do veio do motor, ficando por responder por exemplo quantos pares de pólos tem o motor, qual a frequência de comutação da corrente a que o motor deve operar? Uma vez mais, voltamos a contactar a Moog solicitando mais informações. Uma vez que o interesse da Moog era maior em vender motores novos do que disponibilizar informação sobre os antigos, enviaram apenas documentação relativa aos motores de ultima geração que actualmente vendem. Fazendo

um paralelismo entre os motores antigos e os motores novos conseguimos configurar correctamente os drives.

Para se poder inicializar o drive foi ainda necessário activar os sinais de controlo, numa sequência de sinais específica para desactivar as protecções que o drive tem incorporado por questões de segurança.

Para impedir que o drive active a potência inadvertidamente, o drive tem dois sinais que necessita que sejam activados para desactivar a função de segurança STO (safely switched off torque).

ENPO	ISDSH	STO	Restart inhibit	Controller state	RSH ¹⁾
L	L	on	on	Power stage locked via two channels.	High
H ³⁾	H ³⁾	off	off	Power stage at standby.	Low
(L) ⇒ H ²⁾	(L) ⇒ H ²⁾	off	off	Power stage at standby.	Low
H	(H) ⇒ L	on	on	Power stage locked via two channels.	High
(H) ⇒ L	H	off	on	Power stage locked via one channels.	Low
(L) ⇒ H	H	off	off	Power stage at standby	Low

() previous status
 1) 3x10⁶ Switching cycles at 200 mA (rest position: normally open)
 2) In order to deactivate the restart inhibit the control signals must be simultaneously (ENPO max. errors 5 ms) set to High (H), or ISDSH must be reliably set to High (H) before ENPO.
 3) This only applies, if STO has been disabled by the process described in "2)"

Figura 13 – Activação do drive, manual de utilizador Moog.

Com todas as ligações efectuadas de acordo com a figura 14, avançou-se para a realização dos primeiros testes.

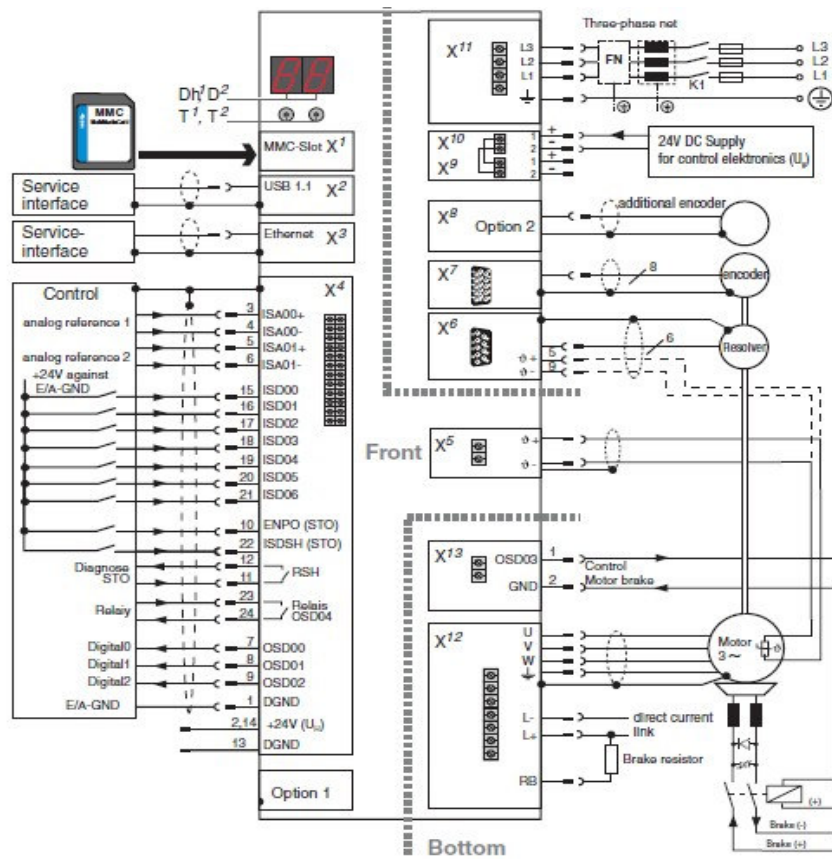


Figura 14 – Esquema de ligações no drive, manual de utilizador Moog.

6.2.1 Teste Inicial

Como teste inicial decidiu-se activar o drive em modo manual, permitindo desta forma averiguar se todas as ligações estavam correctas e se o motor funcionava correctamente. Após activar o modo manual verificou-se que havia várias opções de controlo disponíveis, como por exemplo, controlo de velocidade do motor, controlo de posição do motor, controlo de binário do motor.

Pensando nas possibilidades de aplicação possíveis começou-se por testar o controlo de posição que é fornecido pelos resolvers.

Sendo o resolver um dispositivo que nos indica a posição absoluta do veio do motor, no controlo de posição foi possível indicar ao drive que se pretendia uma determinada posição angular do veio do motor e verificar se este atingia a posição pretendida. A velocidade usada até atingir a posição requerida é um parâmetro pré-configurável dependendo do motor usado e da sua velocidade máxima permitida. Verificou-se também ser possível, em modo manual, obter posições do veio do motor não absolutas mas sim relativas, ou seja, referenciadas à posição anterior atingida.

No modo de controlo de velocidade, verificou-se ser possível predefinir a velocidade pretendida e, após o início do movimento, o controlador acelerava o motor até essa velocidade. A aceleração e a desaceleração são parâmetros configuráveis no drive, sendo posteriormente o drive a gerir ambas, de uma forma autónoma, após serem dadas ordens de arranque e paragem do motor.

Após estes testes manuais, testamos a pergunta fundamental: será possível enviar comandos analógicos para o drive e controlar o motor?

A entrada analógica no drive admite sinais DC variando entre -10 volts a 10 volts. Para se poder proceder ao teste, construiu-se um divisor resistivo de modo a converter os 24 volts, já disponíveis numa saída do drive, para uma gama de 0 a 10 volts aproximadamente, usando um potenciómetro.

Verificou-se que o drive em modo de controlo de posição, não permite realizar o controlo de posição do eixo do motor por sinais analógicos. O drive apenas o permite em modo de teste e não em funcionamento normal. Restava a possibilidade de vermos se era possível fazer o controlo de velocidade com comandos analógicos.

Após executarmos as configurações necessárias do drive, e todas as ligações serem realizadas verificou-se ser possível controlar a velocidade do motor por sinais analógicos. Foi igualmente possível atribuir à entrada analógica uma determinada resolução, ou seja, 0 a 10 volts corresponder, por exemplo, ao intervalo de 0 a 3000 rpm (rotações por minuto).

Verificou-se que o drive tinha também já pré estabelecidos perfis de controlo. Isto é, havia a possibilidade de se escolher entre um perfil trapezoidal ou perfil directo.

O perfil trapezoidal de velocidade consiste numa rampa de aceleração com declive pré definido, estabilização na velocidade máxima e uma rampa de desaceleração até a paragem do motor.

O perfil directo consiste na aplicação directa da ordem proveniente do sinal analógico, ou seja, não era feita nenhuma aceleração, a ordem dada era a ordem aplicada no motor. Caso a diferença entre ordens de velocidade fosse muito grande, o motor respondia com movimentos bruscos e paragens igualmente bruscas, praticamente instantâneas.

Após os testes iniciais, elaborou-se uma possível solução para o problema. Tendo disponível para controlo, o dispositivo de controlo programável Compact-Rio, seria possível obter o ângulo do motor em tempo real? Visto que o drive não permitia enviar o valor da posição do motor para uma saída, a posição teria de ser obtida directamente do resolver para o C-Rio. Caso fosse possível, uma possível solução seria como mostra a figura 15.

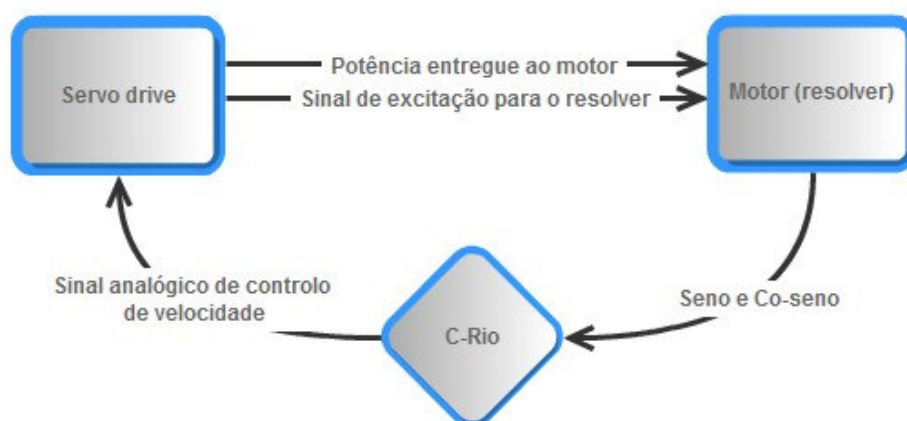


Figura 15 – Diagrama de ligações da solução apresentada

6.3 Compact-Rio

O controlador utilizado neste projecto foi um equipamento da empresa National Instruments designado por Compact-Rio modelo 9002 (figura 16).

Trata-se de uma unidade construída com o objectivo de controlo de sistemas em tempo real possuindo como componentes acessórias um conjunto de cartas de comunicação que permite realizar inúmeras tarefas como por exemplo entradas e saídas analógicas e digitais, ligação a sensores térmicos e ópticos, assim como a equipamento de imagem.



Figura 16 - Compact-Rio

O controlador que utilizamos possui um processador Pentium de 195MHz, 64MB de memória não volátil e 32 MB de memória volátil.

Para comunicação com o exterior, possui uma interface de rede 10BaseT e 100BaseTX Ethernet, compatível com a norma IEEE 802.3, podendo comunicar velocidades de 10Mbps, 100Mbps ou por um outro valor auto negociável.

A ligação entre o controlador e as cartas de comunicação é realizada através de um chassis reconfigurável (Figura 17).



Figura 17 – Chassis reconfigurável

Dentro do chassis reconfigurável existe uma FPGA (field-programmable gate array) programável que é responsável por ler e escrever dados das cartas de comunicação e enviar, se necessário, ao controlador Crio esses mesmos dados através de um barramento PCI (peripheral component interconnect).

Também é capaz de gerar interrupções de modo a manter a sincronização com o controlador, como exemplifica a figura 18.

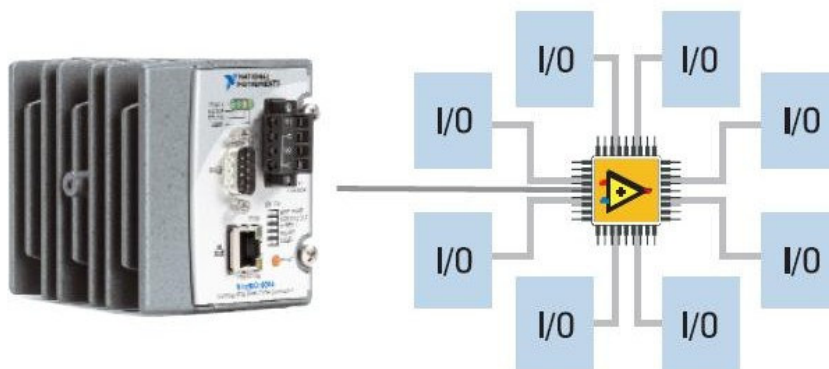


Figura 18 – Esquema de funcionamento da FPGA

As cartas de comunicação são responsáveis pela entrada e saída de sinais fazendo o seu tratamento através de filtros (digitais), conversores analógicos-digitais e digitais-analógicos.

Um dos parâmetros importantes das cartas de comunicação é a sua velocidade de amostragem, algumas são capazes de amostrar a 250 mil amostras por segundo, como por exemplo a carta de comunicação modelo 9205 apresentada na figura 19.



Figura 19 – Exemplo de uma carta de comunicação

Neste projecto em concreto as cartas que utilizamos foram as seguintes: duas cartas responsáveis pelos sinais digitais modelo 9403 e duas cartas responsáveis pelos sinais analógicos modelo 9205 para entradas e modelo 9263 para saídas. A carta de entrada de sinais analógicos possui 32 canais, tendo uma velocidade de amostragem agregada de todos os canais de 250 mil amostras por segundo.

6.4 Programação

6.4.1 Programação em LabVIEW

O equipamento de controlo que utilizamos neste projecto é manufacturado pela firma National Instruments, que é também proprietária da package de programação designada por LabVIEW.

O LabVIEW (acrónimo para Laboratory Virtual Instrument Engineering Workbench) é uma linguagem de programação gráfica, habitualmente designada por linguagem G. Esta linguagem foi desenvolvida pela empresa National Instruments, tendo esta lançado a primeira versão em 1986 para computadores da empresa Apple. Devido à sua facilidade de programação, totalmente transparente a quem tenha os conhecimentos básicos de programação, e a sua elevada versatilidade, o LabVIEW rapidamente se expandiu para outros sistemas operativos tais como o Windows, Linux e Solaris.

Actualmente o LabVIEW é uma linguagem que está especialmente vocacionada para aplicações na área da automação industrial, instrumentação electrónica e controlo. O modelo de programação deste tipo de linguagem, fluxo de dados, confere-lhe vantagens em

aplicações que possuam funções de aquisição de dados e posterior manipulação dos mesmos.

Os programas desenvolvidos em LabVIEW são designados de VI's, Virtual Instruments, devendo-se esta designação ao facto de existirem semelhanças entre os vários blocos constituintes de um VI e os instrumentos, tais como osciloscópios ou multímetros.

Uma aplicação desenvolvida em LabVIEW pode ser constituída por apenas um VI ou por múltiplos VI's. Quando tal acontece é necessário que todos os VI's pertençam ao mesmo projecto sob pena de se verificarem erros durante a execução da aplicação.

Tal como já foi dito anteriormente, todos os programas desenvolvidos possuem o nome de VI, sendo estes constituídos por três elementos: Front Panel; Block Diagram; e Icon.

O Front Panel ou painel frontal representa toda a interface com o utilizador. É aqui que o utilizador interage com a aplicação estando toda a parte de código propriamente dita oculta do utilizador. Um exemplo de uma interface está visível na figura 20.

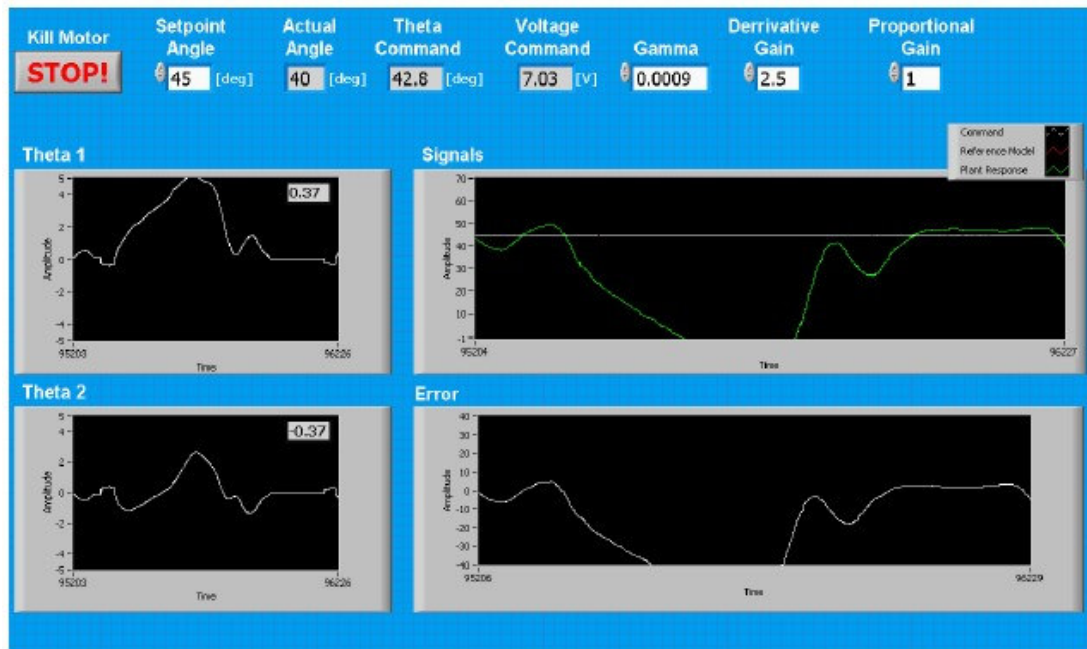
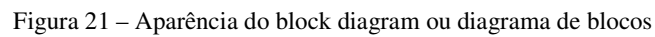


Figura 20 – Exemplo de um front pannel ou painel frontal

O Block Diagram ou diagrama de blocos é o local onde o programador define toda a estrutura do algoritmo, usando para tal pequenos blocos de funções já existentes nas

Numa aplicação, o Block Diagram tem a aparência da figura 21.



31

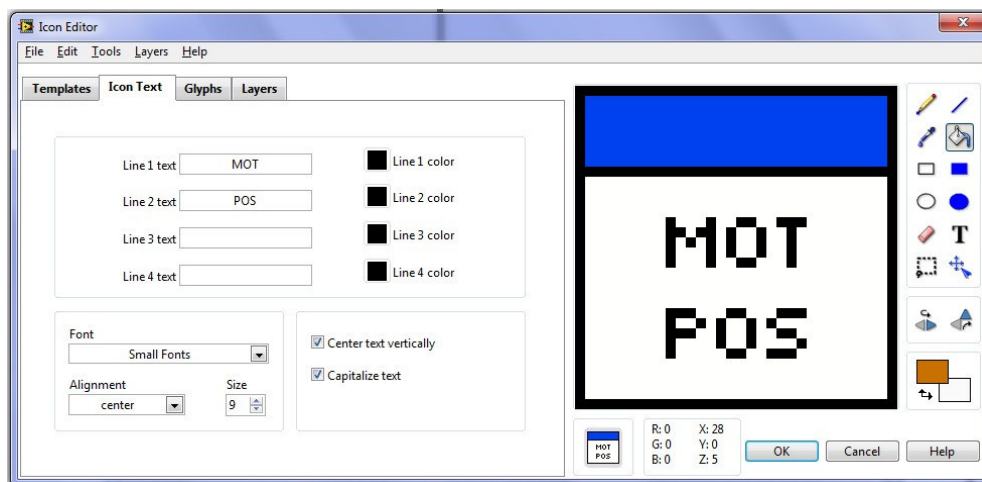


Figura 22 – Ícone de um vi quando usado em outro vi

Tal como anteriormente mencionado, o LabVIEW é uma linguagem de programação gráfica que faz uso de blocos de funções denominados de VI's. Desta maneira, o programador apenas tem que seleccionar os blocos, ou sub VI's, que pretende para a sua aplicação e interligá-los entre si através de linhas. À semelhança de outras linguagens, as sub-rotinas ou no caso do LabVIEW, os vários sub-VI's constituintes de um programa, possuem variáveis de entrada e de saída, sendo estas ligadas entre si da maneira que o programador achar mais conveniente.

A execução de um programa em LabVIEW é baseada em fluxo de dados, ao contrário do que acontece com outras linguagens de programação, o que significa que a chegada de dados a um determinado “nó” determina a ordem da execução de um determinado VI.

Um programador ao construir uma aplicação vai introduzir vários VI's. Desta maneira pode ordenar esses mesmos VI's de forma a ter o fluxo de dados pretendido.

Outro aspecto interessante deste método de programação é o facto de este poder executar múltiplos processos em paralelo.

Uma outra característica bastante interessante nesta linguagem é o facto de muitos dos seus VI's serem polimórficos, ou seja, são capazes de aceitar vários tipos de dados na entrada. Este facto evidencia bastante a versatilidade e a flexibilidade de que esta linguagem de programação possui.

Uma das grandes vantagens desta linguagem de programação é a construção de uma interface gráfica de forma relativamente simples e rápida. O Front Panel

disponibiliza um conjunto de controlos de vários tipos, tais como, botões, interruptores, controlos numéricos ou controlos tipo slide. Apresenta também um leque bastante alargado de indicadores, tais como, tabelas, tráficos e leds. Todos estes indicadores e controlos têm a vantagem de poder ser alterados de modo a apresentarem a informação pretendida da maneira que se achar mais conveniente e prática.

Outra das grandes vantagens que o LabVIEW apresenta é a existência de vários pacotes extra que disponibilizam funções para o controlo do mais variado tipo de dispositivos. Existe por exemplo um pacote para programação de FPGA's, um pacote para controlo de um dispositivo em Tempo Real ou ainda um pacote específico para controlo e simulação.

Nesta dissertação irão ser usados alguns destes pacotes como sejam, o FPGA Module e o Real Time Module.

6.4.2 Controlo de um motor

O primeiro passo na configuração da unidade controladora consistiu na programação da FPGA do chassis reconfigurável. Esta programação é feita utilizando um computador ligado via Ethernet RJ45 à unidade Crio que por sua vez está ligada ao chassis reconfigurável.

A programação inicial da FPGA em LabVIEW consistiu essencialmente no reconhecimento por parte do Crio das cartas de comunicação que introduzimos no chassis.

Após se ter adicionado todas as cartas de comunicação ao projecto, reparou-se que cada entrada e cada saída de uma carta de comunicação, era designada por IO (input output) node.

Como primeiro teste de programação das entradas e saídas de uma carta, colocou-se, no programa em LabVIEW, um ciclo while sem condição de paragem para que a FPGA estivesse em execução indefinidamente. Atribuiu-se em seguida a um nó da carta de comunicação um botão de controlo enviando para o nó o valor true ou false. Em seguida confirmamos, medindo com um multímetro, que a saída estava activa uma vez que apresentava a tensão de zero volts quando o sinal, vindo da FPGA, era falso e de cinco volts quando verdadeiro. Em seguida alteramos o programa de modo a que a saída da carta

só ficasse activa quando uma entrada desta fosse colocada activa, ou seja, também com cinco volts. Para colocar em funcionamento esta rotina do programa foi necessário criar, no Crio, um evento dinâmico que detectasse que uma entrada da carta ficasse activa para activar uma saída. Foi assim necessário criar um programa, a ser executado na FPGA, que apenas lê e escreve os sinais de e para o controlador Crio e monitoriza as entradas e saídas das cartas de comunicação (figura 23).

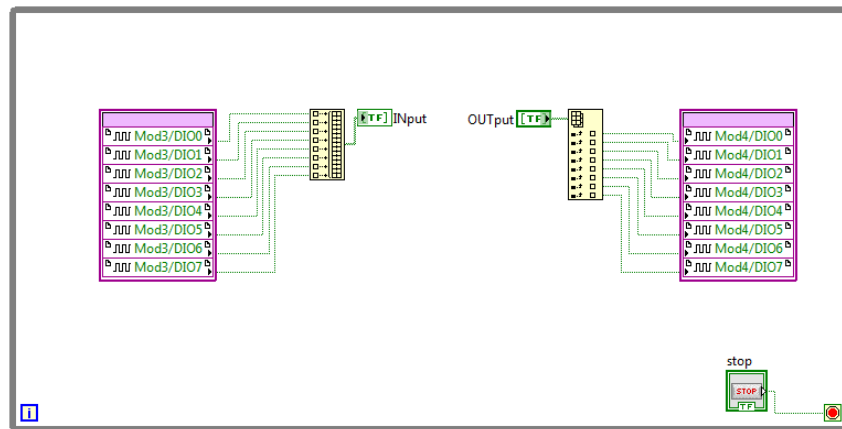


Figura 23 – Programação da FPGA

De seguida criou-se mais um programa para executar no controlador Crio. Este seria responsável por inicializar a FPGA e ficava à espera de um evento que ocorresse. Caso fosse detectado um evento, enviaria para uma porta de saída digital o sinal detectado na porta de entrada digital (figura 24).

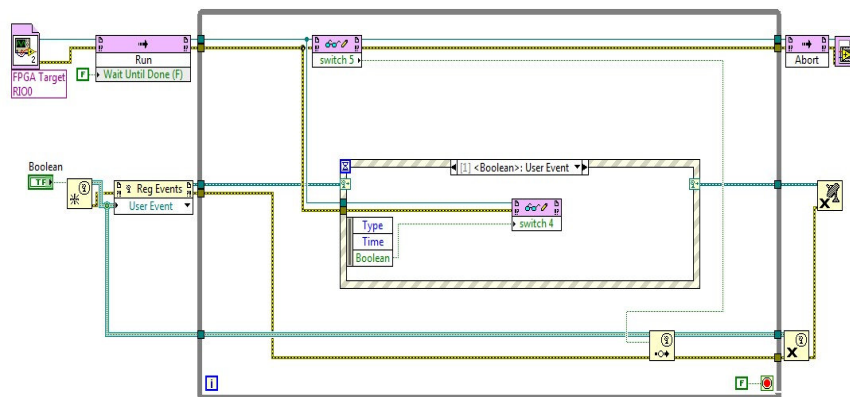


Figura 24 – Estrutura de eventos dinâmicos

Com o correcto funcionamento testado a nível de sinais digitais, partiu-se para a aquisição de um sinal do tipo analógico.

Para a aquisição de um sinal analógico, teria de se acrescentar no programa em execução na FPGA um nó de leitura para cada entrada desejada. Como iria ser necessário adquirir três sinais analógicos, seno, co-seno e sinal de excitação, preparou-se desde o início os nós necessários para essa aquisição. Como primeiro passo, a aquisição seria directa, ou seja, a FPGA adquiria o sinal através da carta de comunicação. Esta já tinha convertido o sinal analógico para digital através de um conversor integrado na própria carta. Então cada amostra que a FPGA adquiria, enviava para o controlador. Como resultado, obteve-se uma onda triangular bastante irregular o que era de esperar uma vez que estamos a amostrar à frequência de 83kHz por entrada, e o sinal de entrada possuía a frequência de 16kHz, logo como frequência insuficiente para obtermos em tempo real uma boa representação do sinal amostrado.

O problema anterior, falta de largura de banda da carta de comunicação não constitui contudo qualquer problema uma vez que estamos interessados apenas na amplitude dos sinais gerados pelo resolver e não no sinal portadora que possui a frequência de excitação do resolver. Temos é que criar um programa detector de pico que recupere a amplitude do sinal à semelhança do que ocorre com a recuperação do sinal de uma onda modulada em amplitude.

Sucedem que os drives Moog que estávamos a utilizar permitem obter uma informação, através do programa de diagnóstico fornecido com os mesmos, que permite visualizar num computador a posição angular do resolver que lhe estiver ligado na respectiva entrada. Tal facilidade não nos foi contudo útil uma vez que não dispúnhamos da possibilidade de fornecer esta informação (posição angular) ao controlador Crio. Perante tal situação tivemos necessidade de desenvolver via LabVIEW o necessário detector de pico.

Para o cálculo da posição angular é então apenas necessário medir os valores de V_{\sin} e V_{\cos} no momento em que o sinal de excitação atinge os máximos. É necessário tomar em atenção para evitar uma possível fonte de erro que se a aquisição das amostras de V_{\sin} e V_{\cos} for realizada ligeiramente desfasada do sinal de excitação podemos ter um erro considerável no cálculo da posição angular. Tendo em vista evitar este problema

construímos em LabVIEW um programa que apenas obtém amostras dos sinais V_{sin} e V_{cos} quando o sinal de excitação é máximo (figura 25).

Como se verifica ainda uma pequena oscilação na posição angular obtida, decidiu-se agrupar as amostras do sinal num conjunto de pontos de modo a ser possível realizar a média destas. Desta forma obteve-se um cálculo mais preciso.

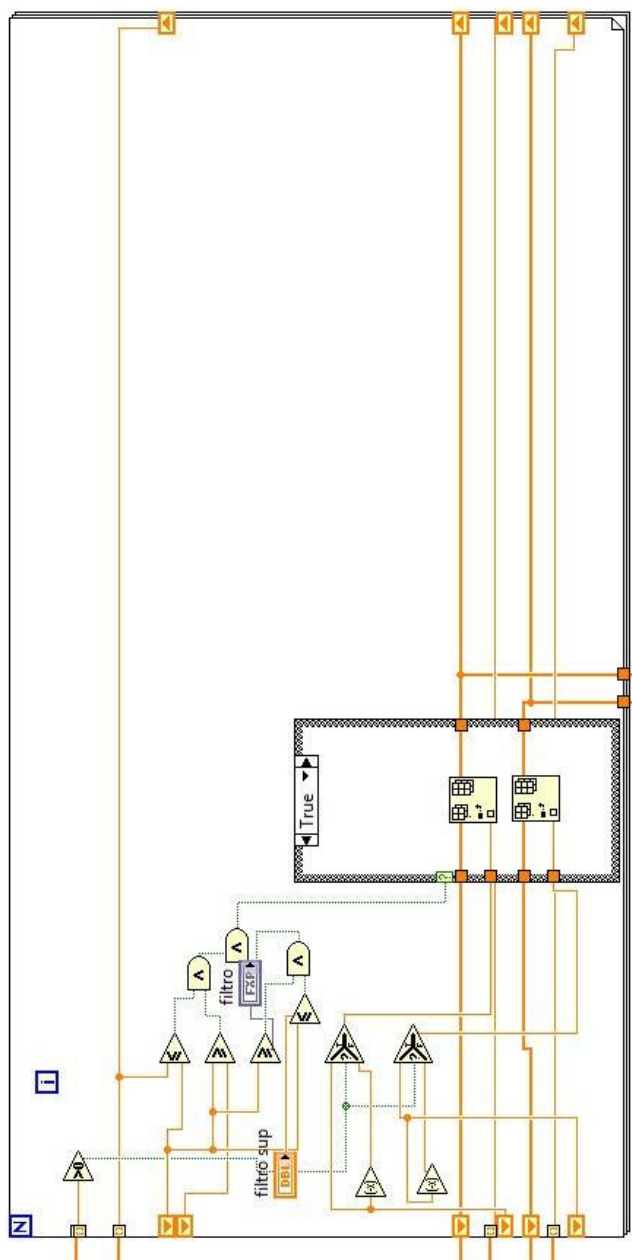


Figura 25 – Detector de máximo e armazenamento de valores em arrays

Deste modo, a aquisição e cálculo da posição angular do veio do motor efectua-se da seguinte forma: a FPGA recebe as amostras dos sinais de excitação, seno e co-seno provenientes da carta de comunicação. Estas amostras são enviadas a um programa residente no controlador Crio que agrupa as amostras num array com um tamanho previamente definido (240 amostras) (figura 26).

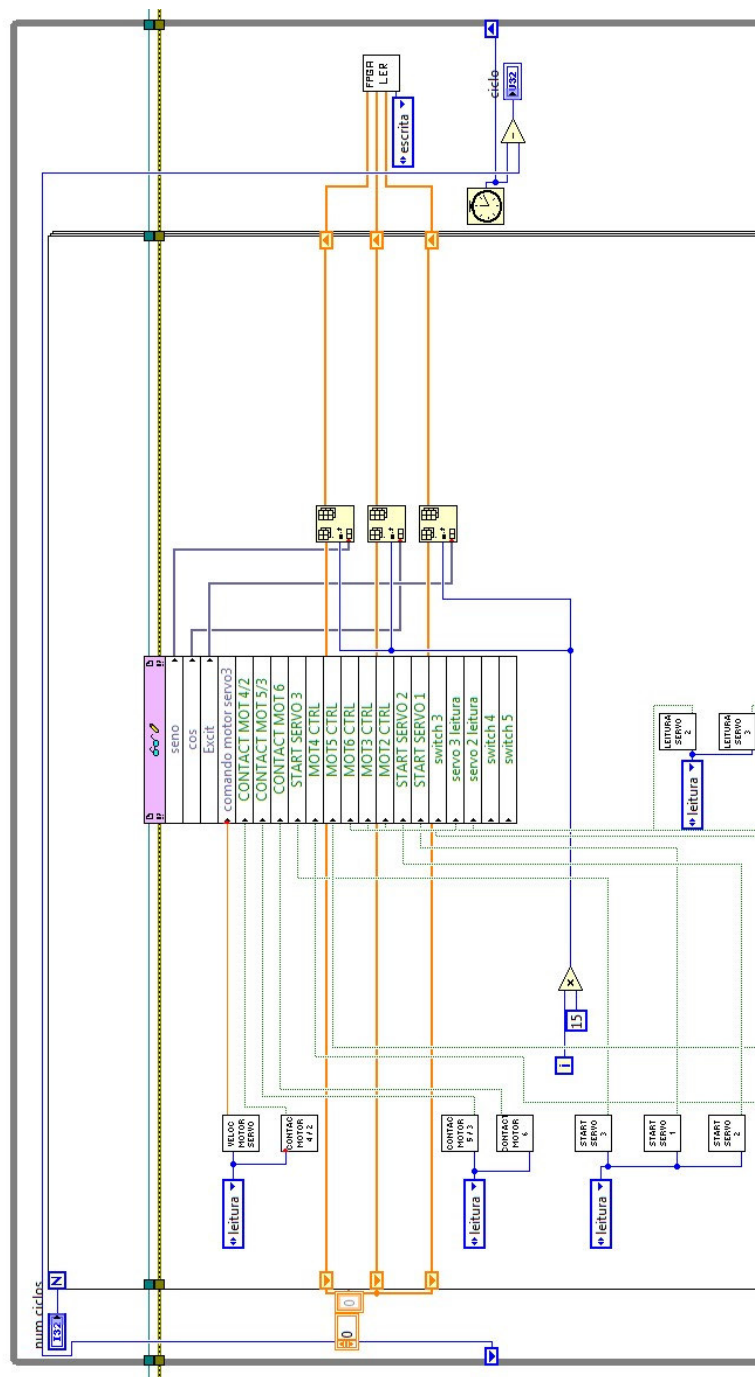


Figura 26 – Comunicação com FPGA

Quando o array atinge o tamanho pretendido (240 amostras), envia esse array para um programa secundário (situado também no Crio) encarregue de calcular a posição angular (figura 27). Nesse programa está presente o detector de pico que detecta os

máximos do sinal de excitação e armazena os valores de Vsin e Vcos correspondentes, criando um novo array (também com 240 amostras) com os mesmos (figura 25).

Com base nos arrays anteriores de Vsin e de Vcos valores de uma determinada posição angular, calculamos a média desses valores. Obtidas as médias, é realizado então o cálculo do ângulo usando a expressão referida aquando da apresentação do resolver (secção 5.2), isto é:

$$\theta = \tan^{-1}\left(\frac{V_{sin}}{V_{cos}}\right)$$

Devido ao facto de necessitarmos que o sinal de excitação atinja o máximo em amplitude para se poder armazenar os valores de Vsin e Vcos correctos, decidiu-se colocar o sinal de excitação adquirido em valor absoluto. Desta forma, duplica-se o número de máximos presentes, aumentando, para o dobro, o número de vezes que se determina a posição angular do veio do motor.

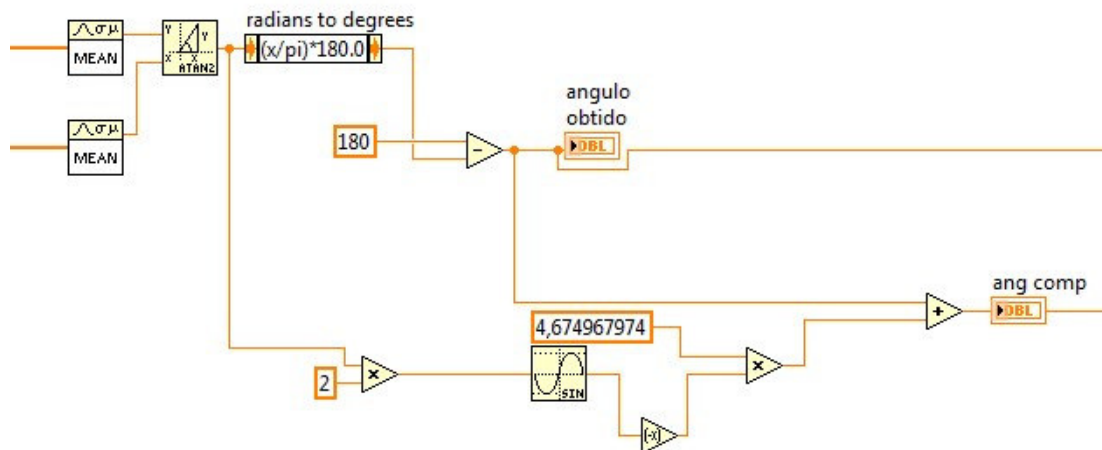


Figura 27 – Cálculo da posição angular

Já temos deste modo a posição angular calculada como se pretendia mas, apenas entre 0 graus e 360 graus.

Como o veio do motor terá de realizar várias voltas até chegar à posição pretendida, torna-se necessário criar um algoritmo que contabilize o número de voltas realizadas

(figura 28), para posteriormente se poder calcular a posição angular. Para tal, criou-se um algoritmo que detectava quando a posição angular passava por zero.

Quando o valor do ângulo transita de 360 para 0 graus, corresponde a uma volta no sentido positivo. Quando este valor transita de 0 para 360 graus, corresponde a uma volta no sentido negativo. Obtendo o número de voltas realizado, é apenas necessário multiplicar o número de voltas por 360 graus e adicionar o resultado à posição angular obtida previamente. Deste modo obtém-se a posição angular contabilizando o número de voltas.

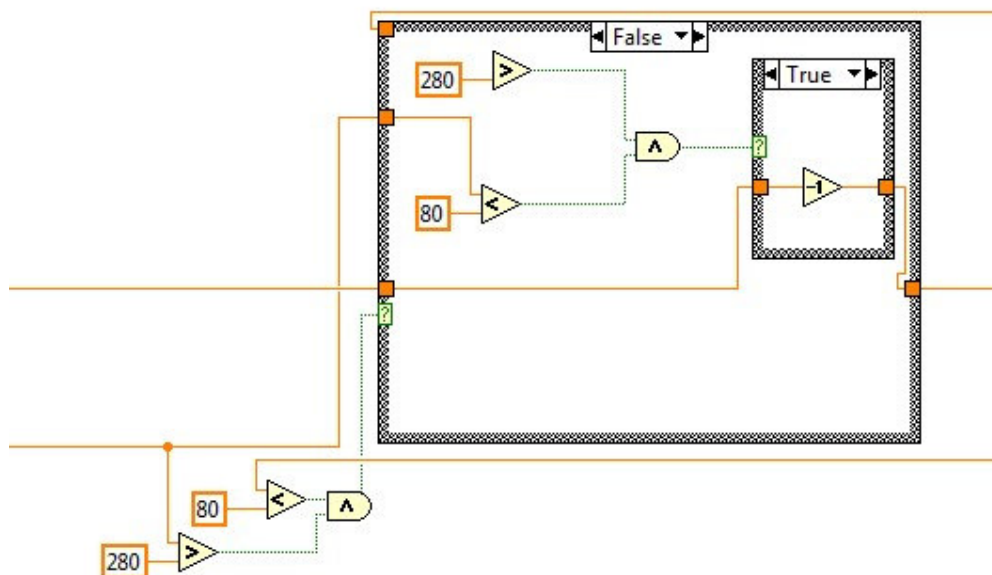


Figura 28 – Detecção de voltas efectuadas

Obtida a posição angular em tempo real, foi necessário programar uma rotina (figura 29) de código responsável pelo controlo do motor.

Como foi referido anteriormente, apenas é possível controlar por sinais analógicos a velocidade do motor. Desta forma, a rotina responsável pelo controlo, terá de monitorizar a posição angular do veio do motor e regular a velocidade do mesmo em função dessa posição angular. Para uma alteração mais fácil de valores para efeitos de teste, decidiu-se realizar o controlo por comparação. O programa irá comparar constantemente se a posição angular actual está compreendida num intervalo pré definido.

Para melhor compreensão o procedimento utilizado foi o seguinte: o utilizador insere no programa a posição pretendida para o veio do motor, o programa compara essa posição pretendida com o valor actual da posição angular do veio do motor, o controlador ordena que o veio do motor comece a mover-se, o programa faz a comparação dos dois valores constantemente averiguando se a diferença entre a posição actual e a posição pretendida se enquadra dentro de um intervalo pré definido pelo programador, caso se encontre dentro do intervalo, a comparação é realizada com uma sequência de intervalos cada vez mais pequenos até se atingir o valor pretendido para a posição angular (erro escolhido de ± 10 graus que corresponde a um erro de posição de aproximadamente de centésimas de milímetro devido à caixa de engrenagens do chassis do robot). O programa implementado reduz a velocidade do motor à medida que a posição angular actual se aproxima da posição pretendida.

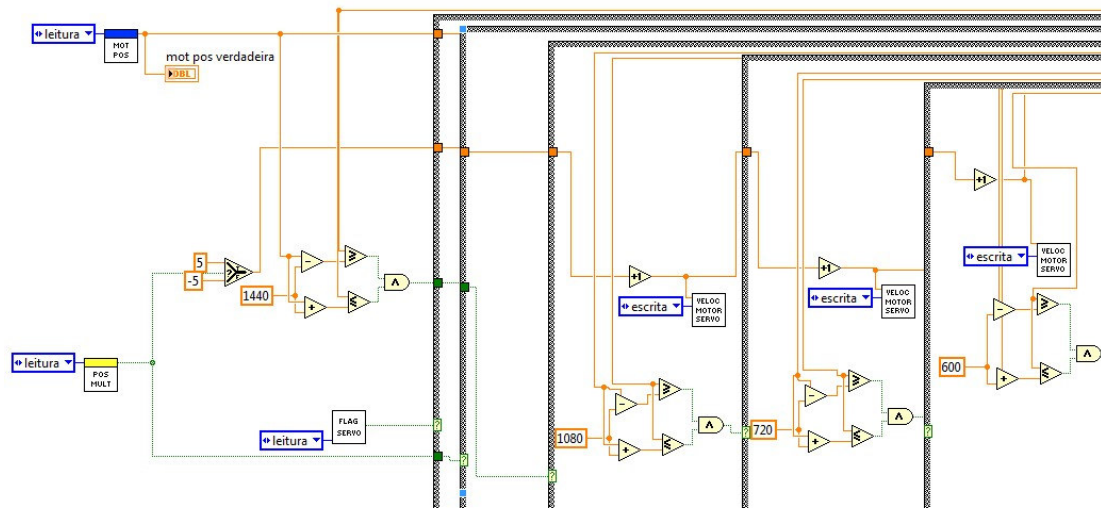


Figura 29 – Controlo da velocidade do motor

Nesta fase, é necessário introduzir mais um conceito, o qual se designa por função global.

Em linguagem LabVIEW, uma função global consiste num ciclo while que apenas se executa uma vez sempre que é chamado (figura 30). Dentro do ciclo while existe uma estrutura case que, em função da operação desejada indicada pelo programador, selecciona o caso desejado, podendo corresponder a leitura, escrita ou inicialização de valores. Esta

função é de extrema importância, pois desta forma torna-se possível guardar valores que serão usados por outras rotinas ou outros programas em execução.

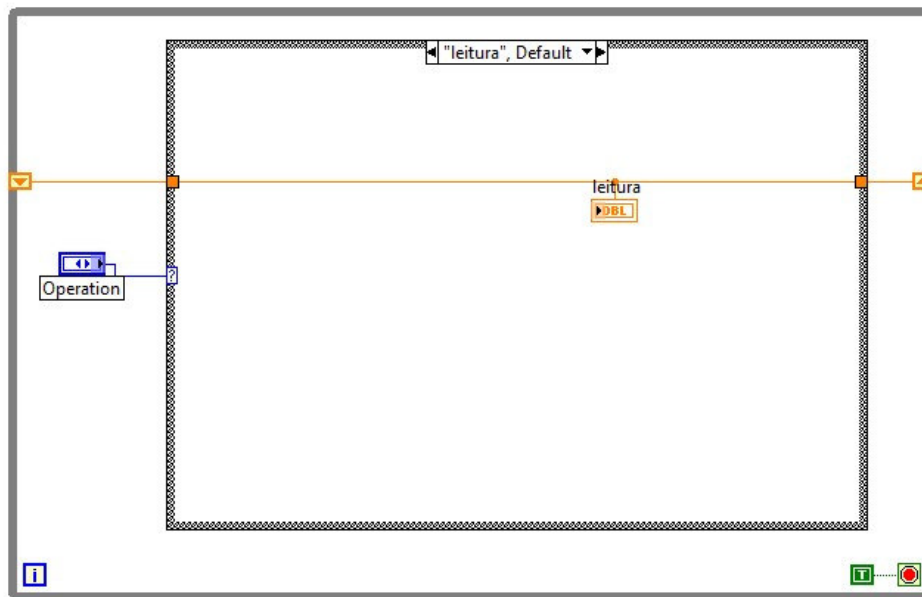


Figura 30 – Função global

Para envio do comando de velocidade do motor foi usada uma função global. Sempre que a comparação entre a posição angular actual e a posição pretendida era efectuada, é guardado um novo valor na função global de velocidade. A saída analógica, encarregue de enviar o sinal de controlo de velocidade para o drive, envia o sinal em amplitude igual ao valor guardado na função global, dentro do intervalo permitido pela carta de comunicação, -10 a 10 volts. No drive Moog, foi definido previamente (utilizando o software disponibilizado pela Moog para configurar o drive) durante a pré configuração do drive, a resolução desse sinal, ou seja, quantas rotações por minuto corresponderiam a 10 volts aplicados na entrada analógica.

Devido a verificarem-se oscilações nos sinais V_{sin} e V_{cos} (originadas pelo ruído propagado proveniente do drive sempre que se activava a potência fornecida ao motor) no cálculo da posição angular também se verificavam essas oscilações. Desta forma, quando o motor atingia a posição angular pretendida, após alguns segundos, o valor da posição angular calculada oscilava, desviando-se do valor real, fazendo com que o controlador

ajustasse a posição novamente. Observou-se que o controlador permanecia a ajustar a posição do motor até ordem manual de paragem. Este fenómeno mostrou a necessidade de haver um melhor tratamento do sinal proveniente do resolver, para um cálculo mais estável da posição angular que vamos passar a descrever.

Devido ao facto de haver pouca memória disponível no controlador, e o esforço computacional ser muito elevado aumentando o tempo de execução de cada ciclo, não foi possível introduzir técnicas avançadas de processamento de sinal. A única solução possível de ser aplicada seria a introdução de filtros na recepção dos sinais analógicos provenientes do resolver (V_{sin} e V_{cos}) e do drive (sinal de excitação).

Como primeiro passo, aplicou-se um filtro passa baixo por software. Para se proceder à correcta configuração do filtro, foi necessário saber qual a frequência exacta de amostragem. Usando o controlo de tempo de execução da FPGA, determinou-se que a máxima frequência de amostragem era de 40kHz. Devido ao ruído estar muito próximo da frequência do nosso sinal, o filtro passa baixo não efectuava uma boa filtragem deixando passar uma quantidade de ruído considerável. Um filtro passa banda foi aplicado em substituição do filtro anterior. Usando um filtro passa banda, atribuiu-se inicialmente uma banda de passagem de 2kHz entre 15kHz e 17kHz. Foi-se diminuindo este intervalo de passagem até se determinar qual a banda de passagem óptima para o sinal. Constatou-se que quanto menor a banda de passagem, mais lento era a resposta do filtro, afectando o tempo de cálculo da posição angular. Tentando encontrar um equilíbrio, um compromisso entre rapidez e ruído, determinou-se que a melhor banda de passagem se encontrava entre os 16060Hz e os 16160Hz, havendo uma banda de passagem de 100Hz. Analisando um pouco mais os resultados, concluiu-se que ainda se verificava uma oscilação na posição angular calculada mas muito inferior à que tínhamos anteriormente pelo que o filtro melhorava os resultados.

Teimando reduzir o erro de posição obtido, efectuamos um estudo no qual consistiu em anotar para cada grau, dez medidas da posição angular calculada. Fazendo este procedimento desde os 0 graus até aos 360 graus, foi possível calcular uma média das medidas efectuadas e calcular o desvio. Por desvio entende-se a diferença entre a média de

valores anotados e o valor da posição angular lida no software de diagnóstico do drive. Constatou-se que a posição angular calculada era bastante precisa para determinadas posições do veio do motor e a mesma posição apresentava um desvio de aproximadamente ± 10 graus para outras posições do veio do motor.

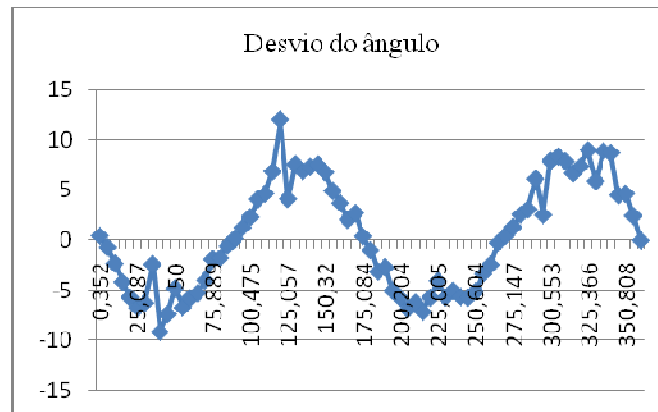


Figura 31 – Desvio obtido no cálculo do ângulo

Como podemos ver na figura 31, o desvio era próximo de zero para as posições 0°, 90°, 180°, 270° e 360° (isto é, quando um dos sinais do resolver passava por zero). O mesmo, tinha valor máximo nas posições 45°, 135°, 225° e 315° (quando ambos os sinais do resolver estavam longe do zero).

Tendo em conta a forma de onda obtida, aplicou-se uma compensação de modo a atenuar este desvio. A compensação consistiu na criação da forma de onda visualizada na figura 31, usando a função $-D \sin(2 * \alpha)$, sendo D a média do desvio do ângulo e α a posição angular calculada. Aplicando filtragem e compensação conseguiu-se um controlo do motor com um erro de mais ou menos cinco graus.

À primeira vista aparenta ser um erro considerável sendo o manipulador usado para trabalhos de precisão, mas tendo em conta que na parte mecânica do manipulador existe uma engrenagem de desmultiplicação, este erro apenas será da ordem de uma décima de milímetro no caso dos eixos com uma desmultiplicação de 50 e da ordem de dez centésimas de milímetros no caso de uma desmultiplicação de 100.

Tendo-se verificado que o processador do controlador (Pentium a 195MHz) não permitia ciclos de execução suficientemente rápidos para o controlo de velocidades muito

elevadas dos motores, pois perdia-se o controlo dos mesmo por não ser possível medir os ângulos (fornecidos pelo resolver através do Crio) em tempo útil, foi necessário calcular a velocidade máxima que podíamos especificar para os drives.

Possuindo a FPGA o tempo de execução de um ciclo da é de $24\mu s$, verificou-se que o tempo de execução da rotina, responsável pela leitura dos valores provenientes da FPGA e agrupando-os em conjuntos de pontos num array oscila entre 2 ms. e um máximo de 20 ms, este último corresponde ao momento em que o controlador reorganiza a memória volátil.

O tempo de execução de um ciclo do programa principal responsável pelo cálculo do ângulo e controlo da movimentação do veio do motor é de 80 ms, pelo que no pior cenário, o tempo de execução total será de 100 ms. Este valor corresponde a uma frequência de 10Hz.

Colocando o veio do motor a deslocar-se uma volta por segundo, o controlador apenas consegue determinar a posição angular dez vezes numa volta de 360 graus. Isto significa que a posição angular será calculada de 36 em 36 graus. Caso o veio do motor se desloque a uma velocidade de duas voltas por segundo, o controlador apenas calcula a posição angular 5 vezes durante uma volta 360 graus. Isto significa, apenas calcular a posição angular de 72 em 72 graus.

Não tendo sido possível, por motivos económicos da firma, adquirir um controlador mais rápido, tivemos que decidir predefinir a velocidade máxima do veio do motor a uma volta por segundo.

O problema de falta de velocidade foi minimizado usando o método de aproximação lenta que já descrevemos. Assim, a posição angular actual é comparada a um intervalo onde o valor pretendido se encontra no meio desse intervalo. À medida que o veio do motor se aproxima da posição pretendida, a comparação é feita com intervalos cada vez mais pequenos, e diminuámos a velocidade do veio do motor de modo a podermos aumentar o número de leituras da posição angular por volta. Desta forma, o número de vezes que a posição angular é calculada vai aumentando, aumentando assim um pouco a precisão, diminuindo o erro introduzido pela baixa velocidade de cálculo do controlador.

Tendo em vista calcularmos a desmultiplicação introduzida pelas engrenagens acopladas aos motores de cada eixo, procedemos do seguinte modo.

Utilizando o perímetro de uma circunferência $2\pi r$, onde r representa o raio da circunferência, é possível determinar qual o ângulo que corresponde ao movimento realizado pelo eixo do manipulador. Tendo esse valor e sabendo rotação efectuada pelo veio do motor, obtemos a desmultiplicação do eixo do manipulador.

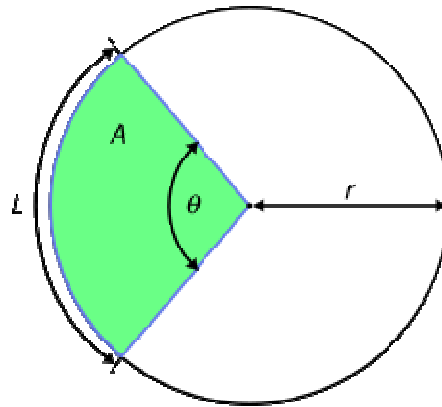


Figura 32 – Deslocação angular

Visualizando a figura 32, o procedimento utilizado foi o seguinte: mediu-se o raio da circunferência que o eixo do manipulador descreve, sinalizou-se a posição inicial quer do eixo do manipulador quer do veio do motor e colocou-se o motor em funcionamento alguns segundos. Após a paragem do motor, anotou-se a posição final do eixo do manipulador e do veio do motor obtendo-se a diferença entre o valor inicial e o valor final. Com o valor da deslocação do manipulador (arco L na figura 32), calcula-se o ângulo θ usando a expressão $\theta = \frac{L \cdot 360}{2 \cdot \pi \cdot r}$. Com o valor da deslocação angular do veio do motor $D\theta$, obtém-se a desmultiplicação $\frac{D\theta}{\theta}$.

Em todos os eixos do manipulador, a desmultiplicação obtida foi da ordem das dezenas à excepção do veio do pulso em que atingia a ordem das centenas, verificando-se que um erro de 5 graus no veio do motor corresponde a um erro na ordem das décimas de milímetro no eixo do manipulador à excepção do pulso em que é da ordem de dez centésimas.

6.4.3 Controlo de vários motores com o mesmo drive

Até este momento, apenas se efectuou o controlo de um motor. Para se proceder ao controlo de vários motores utilizando o mesmo drive, foi necessário introduzir modificações tanto ao nível de software como ao nível de ligações físicas.

As ligações, até ao momento, consistiam em ligar directamente a potência do drive para o motor, o drive ao resolver do motor, o resolver ao controlador Crio e o drive também ao controlador Crio, como se encontra exemplificado na figura 15. Agora tornou-se necessário interligar o drive a vários motores, assim como o resolver de cada motor ao drive e ao controlador, como indica a figura 33.

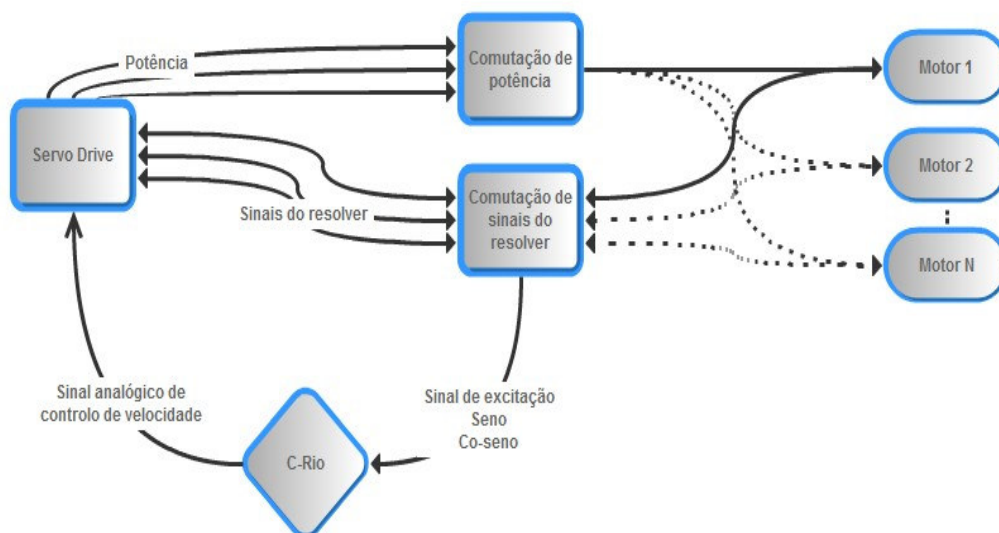


Figura 33 – Diagrama de ligações

Relembrando que a solução para o problema consistia em utilizar os três drives disponíveis para realizar três movimentos simultâneos, seria necessário que o controlador fosse capaz de amostrar os sinais de controlo (seno, co-seno, excitação do resolver) dos três drives em simultâneo. Seriam nove sinais, em nove entradas analógicas simultaneamente. Infelizmente constatou-se que o tempo de execução de um ciclo da FPGA aumentava drasticamente para $72\mu\text{s}$, ou seja, a frequência de amostragem seria 13,8kHz, isto é, abaixo da frequência do sinal de excitação do resolver (16kHz). Este problema resulta do facto de a velocidade de amostragem da carta de comunicação ser agregada, ou seja, a velocidade é dividida pelo número de entradas utilizadas. Deste modo,

usando o controlador disponível, apenas é possível controlar um drive de cada vez, ou seja, teremos apenas um movimento de um eixo de cada vez, ficando a velocidade de amostragem em 40kHz correspondendo a um ciclo de execução de 25 μ s.

Para efectuar a comutação de potência, utilizamos um circuito eléctrico possuindo contactores. Estes dispositivos electromecânicos são constituídos por uma bobine, que produz um campo magnético atraindo uma parte móvel ligada a um conjunto de contactos. O movimento da parte móvel provoca o fecho ou abertura dos contactos do contactor. Visto que os motores utilizados são trifásicos, utilizaram-se contactores de 3 contactos.

Os contactores são accionados por um sinal de corrente alternada de 230 volts monofásico, havendo também a possibilidade de serem accionados com sinal de 24 volts de corrente contínua, caso estejam preparados para o efeito.

Para a comutação dos sinais de controlo, isto é, os sinais provenientes do resolver e do drive, foram utilizados relés de baixa potência. Estes dispositivos têm um modo de funcionamento semelhante aos contactores, sendo uma corrente contínua que circula na sua bobine criando um campo magnético atraindo uma parte móvel, provocando a abertura ou fecho dos contactos. Têm a vantagem de o sinal de accionamento ser de baixa voltagem, normalmente compreendido entre 5 e 24 volts, isolando galvanicamente o circuito de controlo do circuito secundário onde podem circular sinais mais elevados em tensão.

Sendo oito sinais provenientes de cada motor, os quais são: seno, co-seno, sinal de excitação, sinal do sensor de temperatura e as respectivas referências para cada sinal, agrupou-se os relés em grupos de oito. Dentro de um grupo de relés, estes estão ligados entre si de modo a se ligarem ao mesmo tempo quando accionados.

Para accionamento dos grupos de relés, são utilizadas as saídas digitais do controlador Crio, as quais fornecem 5 volts na saída.

Para accionamento dos contactores foi também necessário utilizar um relé de 5 volts, o qual controlava o sinal de corrente alternada monofásico que acciona os contactores.

Estas alterações nas ligações eléctricas obrigaram a alterações no software de modo a garantir a comutação dos sinais de controlo e de potência entre o drive e controlador e os motores e respectivos resolvers.

Cada saída digital terá que ser controlada por software. Um factor importante é a sincronização. É necessário que a comutação seja no momento certo para evitar erros, ou um funcionamento descontrolado do manipulador.

Para implementação da comutação utilizou-se o seguinte esquema. Existindo um ciclo while responsável pelo tratamento dos sinais, cálculo da posição angular e controlo da velocidade do veio do motor, criou-se um novo ciclo while, que se executa em paralelo com o ciclo anterior, controlando a comutação dos sinais, activação e desactivação do drive, controlo de todas as flags de sinalização implementadas, ordem de paragem e arranque do motor.

Verificou-se que a comutação do sinal proveniente do resolver, enquanto o drive estava activo, isto é, enquanto enviava potência para o motor, desencadeava um alarme no drive, ficando bloqueado a aguardar intervenção do utilizador. Para tal não suceder, tornou-se necessário assegurar que o drive seja colocado em modo de espera, no qual não providencia potência para o motor, antes que o sinal proveniente do resolver seja comutado para outro motor. Todo o controlo dos relés e dos contactores foi implementado usando funções globais do tipo boolean.

Também foi necessário ter em conta que, antes de se poder activar um contactor e um grupo de relés correspondentes a um determinado motor, é preciso desactivar previamente o contactor e o grupo de relés usados no motor anterior. Para garantir esta sincronização foi usado um conjunto de estruturas temporais. Estas garantem que determinados conjuntos de código apenas irão executar em circunstâncias específicas, tornando o código robusto, evitando erros de funcionamento.

Para cada conjunto de motores, ou seja, para cada drive, foi criada uma estrutura case que selecciona o motor que se vai utilizar através de um controlo manual como se pode ver na figura 34.

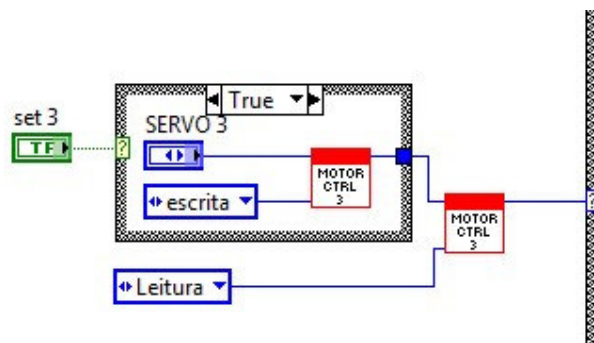


Figura 34 – Controlo manual

Para uma maior segurança, aplicou-se um controlo manual designado set3. Este controlo impede a escrita indevida na função global, responsável pela selecção do motor em funcionamento. A função global apenas assume o valor que se pretende escrever, quando o utilizador valida no controlo manual. Dentro da estrutura case, aplicou-se um ciclo while que fica em execução até o veio do motor atingir a posição angular pretendida. Desta forma, é assegurado que as flags de controlo permanecem inalteráveis durante o movimento do motor. Dentro do ciclo while foi aplicada uma estrutura sequencial. Esta estrutura executa o código desenvolvido no seu interior de uma forma sequencial garantindo a correcta sequência de activação das flags de controlo. Deste modo estabelecemos quatros tarefas que se tornam necessárias executar sequencialmente para se proceder à comutação entre os eixos que o mesmo drive vai accionar. Estas são:

Como podemos ver na figura 35, a primeira tarefa consiste em garantir que as flags de controlo de todos os motores, excepto o motor que se irá utilizar, se encontrem desactivadas para a correcta comutação dos sinais.

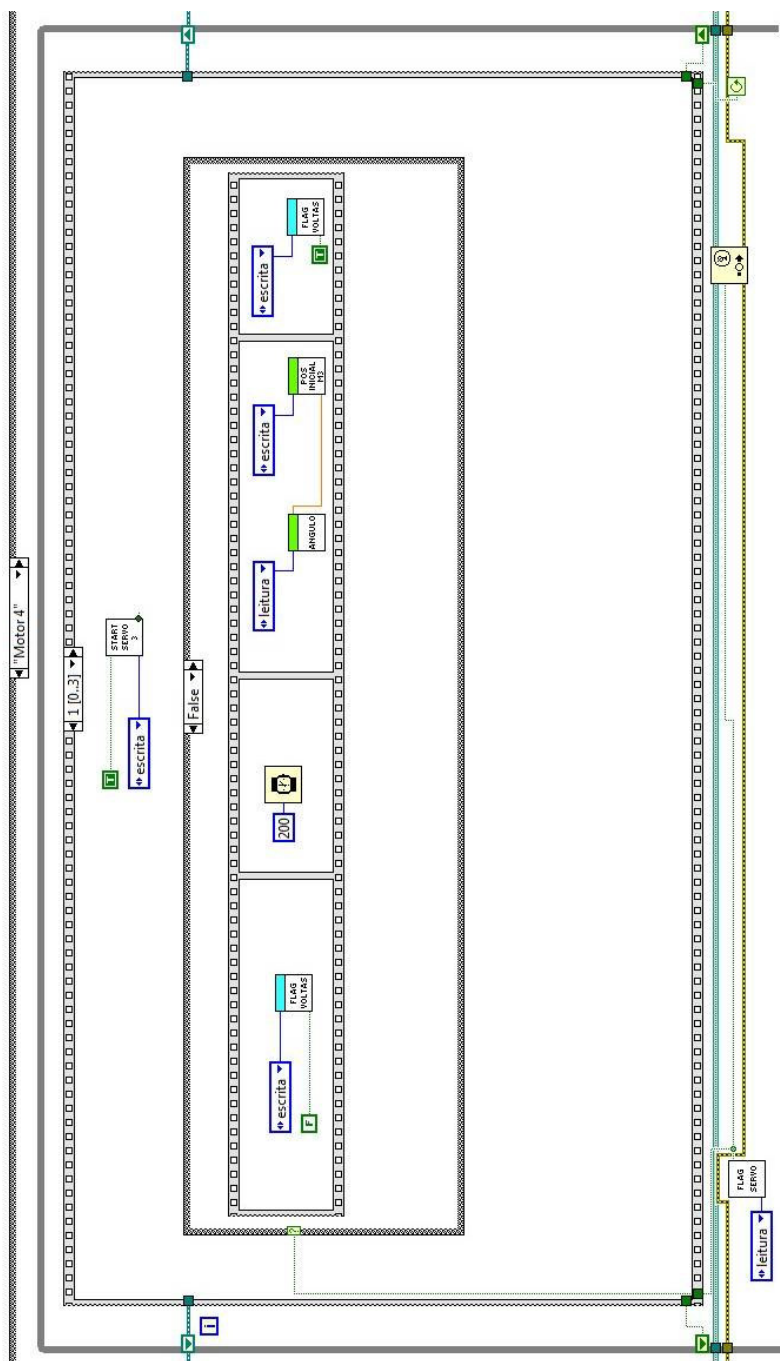


Figura 36 – Segunda tarefa da estrutura temporal

A segunda tarefa (figura 36) consiste em garantir que o contador de voltas se encontra a zero para a correcta leitura da posição angular inicial. Neste modo de funcionamento, a referência não é absoluta mas sim relativa, ou seja, o veio do motor desloca-se a partir da posição inicial até à posição pretendida. Esta tarefa também activa a flag de controlo do drive designada por start servo, passando do estado de espera para o estado activo enviando potência para o motor. O drive foi configurado de modo a desactivar o travão do motor quando transita para o estado activo.

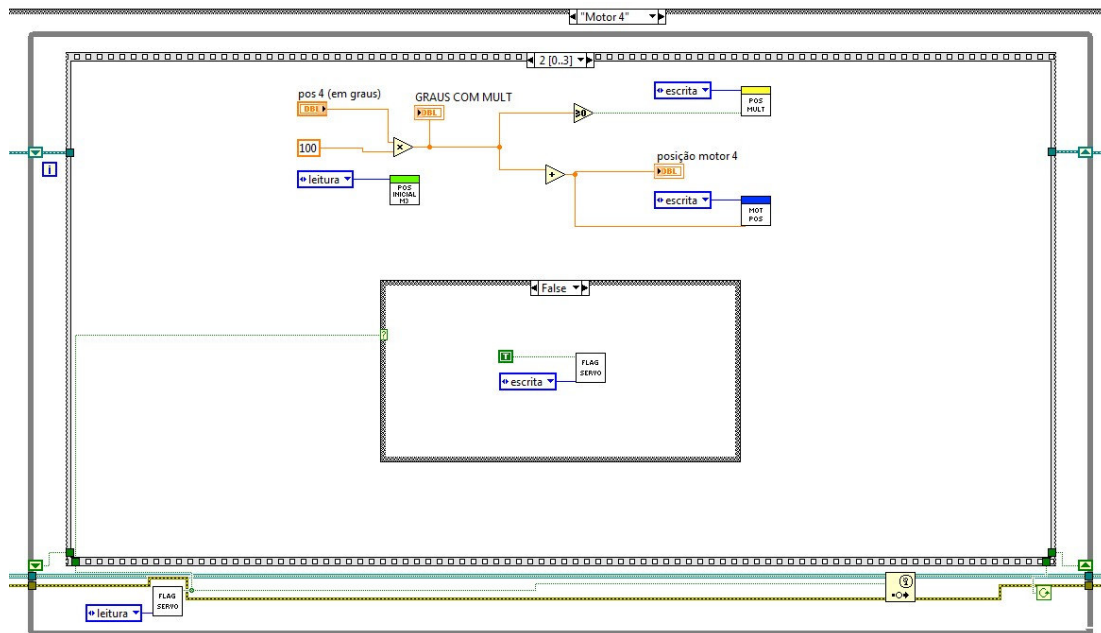


Figura 37 – Terceira tarefa da estrutura temporal

Numa terceira tarefa (figura 37) é efectuado o cálculo da posição angular pretendida. O valor introduzido pelo utilizador é multiplicado pela razão de desmultiplicação do manipulador, é somado à posição inicial anteriormente estabelecida e é enviado para a função global mot pos, a qual é responsável pelo armazenamento da posição pretendida. Essa função global irá ser lida pela rotina de controlo de velocidade do motor no ciclo while em execução paralela. É também activada a flag de inicio de movimento do drive designada por flag servo, que permite a movimentação do veio do motor, funcionando como um sinal de inicio de movimento. Caso a posição angular pretendida seja positiva, o veio do motor rodará num sentido, caso seja negativa rodará no outro sentido.

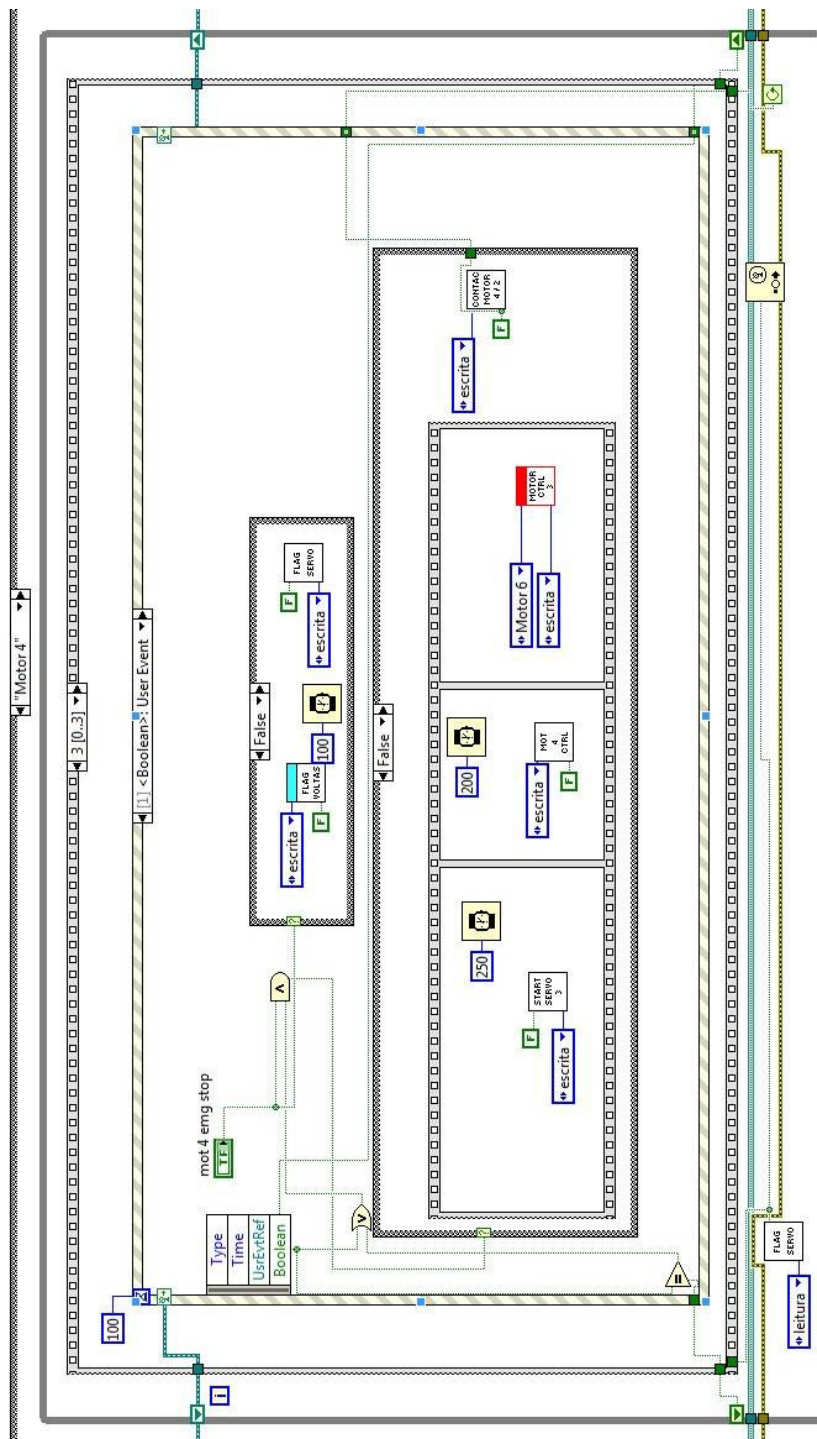


Figura 38 – Quarta tarefa da estrutura temporal

A quarta tarefa (figura 38) consiste numa estrutura de eventos dinâmicos, a qual aguarda que a flag de controlo do drive seja desactivada indicando que o motor chegou à posição pretendida.

Para maior robustez do código, aplicou-se um filtro o qual verifica se houve uma transição de estado activo para inactivo da flag de controlo, rejeitando todas as outras transições que possam ocorrer. Quando o motor atinge a posição pretendida, é realizada a comutação do sinal de potência assim como dos sinais de controlo, o drive é colocado em modo de espera e é inicializado o controlo do motor seguinte. O procedimento repete-se até todos os motores atingirem a posição indicada manualmente pelo utilizador (ou no futuro por outro equipamento). Designou-se por modo “manual” esta execução do programa.

A interface gráfica para uso do utilizador encontra-se na figura 39.

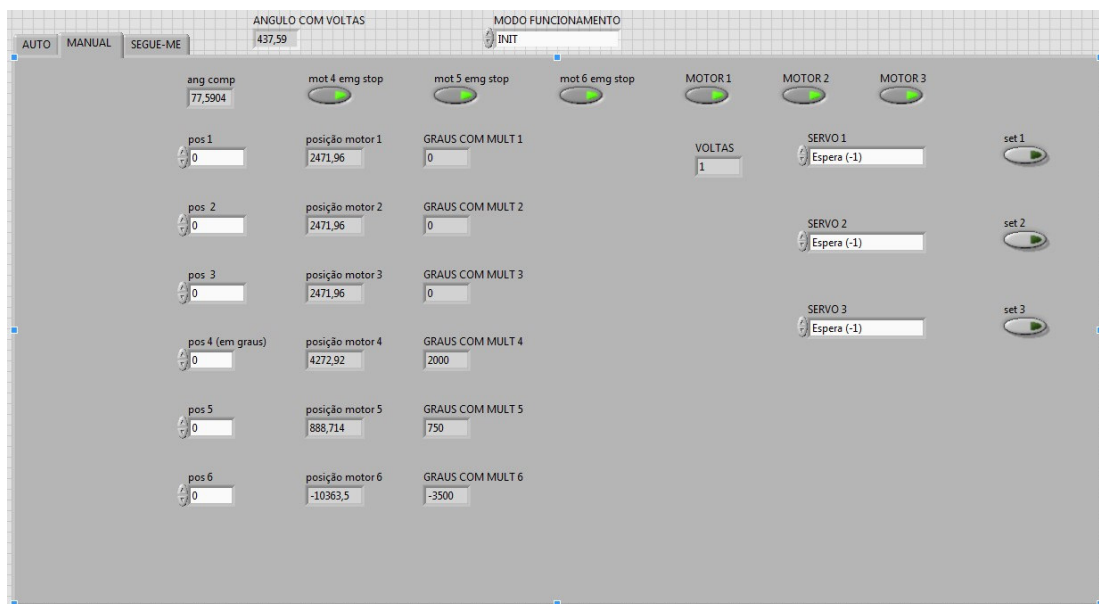


Figura 39 – Interface com o utilizador em modo manual

6.5 Modos de funcionamento

Para exemplificação das enumeras possibilidades de funcionamento de um manipulador industrial, criamos três modos de funcionamento. O primeiro modo de funcionamento foi já demonstrado anteriormente.

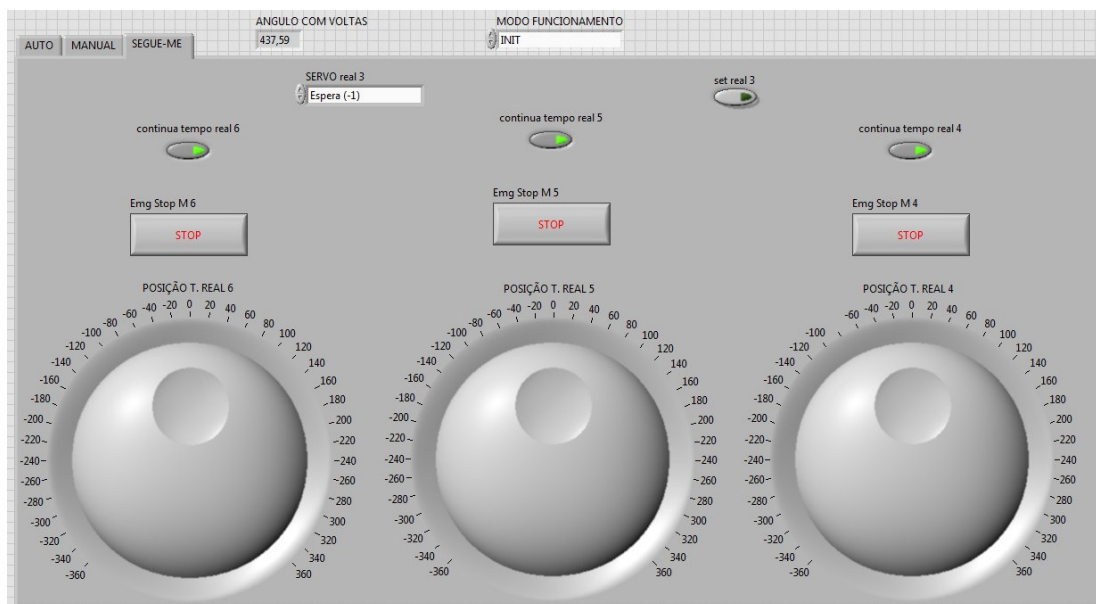


Figura 40 – Interface com o utilizador em modo “segue-me”

O segundo modo de funcionamento foi designado de “segue-me” (figura 40). Neste modo de funcionamento, o utilizador pode seleccionar o motor que pretende usar e através de uma interface gráfica, colocar o motor em funcionamento. Sendo a interface um botão rotativo, o motor segue o movimento desse mesmo botão.

Para se implementar este modo de funcionamento foi necessário alterar a activação das flags de controlo assim como o cálculo e validação da posição angular. Toda a estrutura usada no modo de funcionamento manual foi mantida, tendo apenas sido alterado o código no seu interior. Verifica-se que a primeira e segunda tarefa permanecem iguais ao modo de funcionamento manual. Em primeiro lugar comuta-se os sinais de potência e de controlo, obtém-se a posição inicial e activa-se o drive.

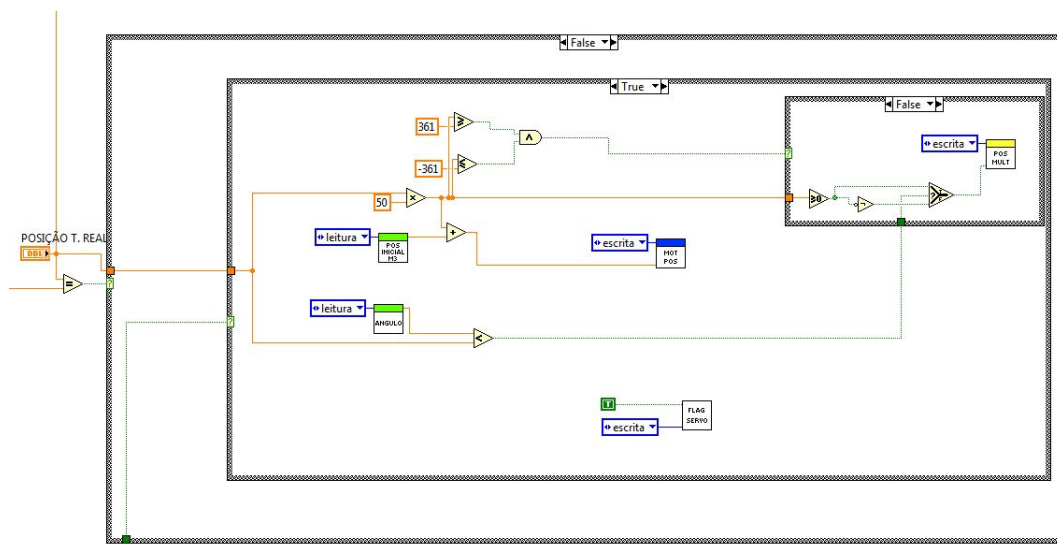


Figura 41 – Terceira tarefa do modo “segue-me”

No entanto, na terceira tarefa (figura 41), aplicou-se um filtro de modo a rejeitar posições angulares iguais, ou seja, enquanto o posição angular pretendida for igual durante vários ciclos de execução, o motor permanece nessa posição. Quando a posição angular pretendida se alterar há duas hipóteses: a primeira hipótese reflecte a situação em que o motor se encontra parado, sendo então calculada a posição angular pretendida, como referido no modo manual de funcionamento; a segunda hipótese reflecte a situação em que o motor já se encontra em movimento. Nesta situação, o valor da posição angular pretendido é actualizado e é efectuada a comparação entre o valor pretendido e o valor da posição angular em que se encontra.

Caso o novo valor seja menor, é indicado à rotina responsável pelo controlo de velocidade do veio do motor para inverter o sentido de rotação. Esta indicação é realizada através da função global “pos mult”.

Na quarta tarefa, o funcionamento é semelhante ao modo de funcionamento manual com uma diferença. Por questões de segurança, foi aplicado um botão de paragem de emergência, dando a possibilidade ao utilizador de para o movimento sempre que queira. Neste modo de funcionamento, a referenciação da posição é absoluta, sendo a posição angular inicial sempre preservada pois o número de voltas realizado pelo veio do motor nunca é forçado a zero após o início de movimento.

Como terceiro modo de funcionamento, desenvolveu-se uma rotina que coloca o manipulador a realizar uma tarefa repetitiva. Para tal, foi necessário efectuar grandes alterações no modo de referênciação. Ao contrário do modo de funcionamento anterior no qual se controlava um motor de cada vez, neste modo não há apenas um motor envolvido mas sim vários. Torna-se então necessário armazenar todas as posições iniciais de todos os motores e armazenar as deslocações efectuadas pelos mesmos.

Estruturalmente, a rotina é semelhante às rotinas anteriores, mas o seu conteúdo foi alterado. Na primeira tarefa é realizada comutação dos sinais de potência e dos sinais de controlo para o motor pretendido. Na segunda tarefa (figura 42) da estrutura sequencial, foi implementada uma função global, a qual funciona como um detector de transição de estado.

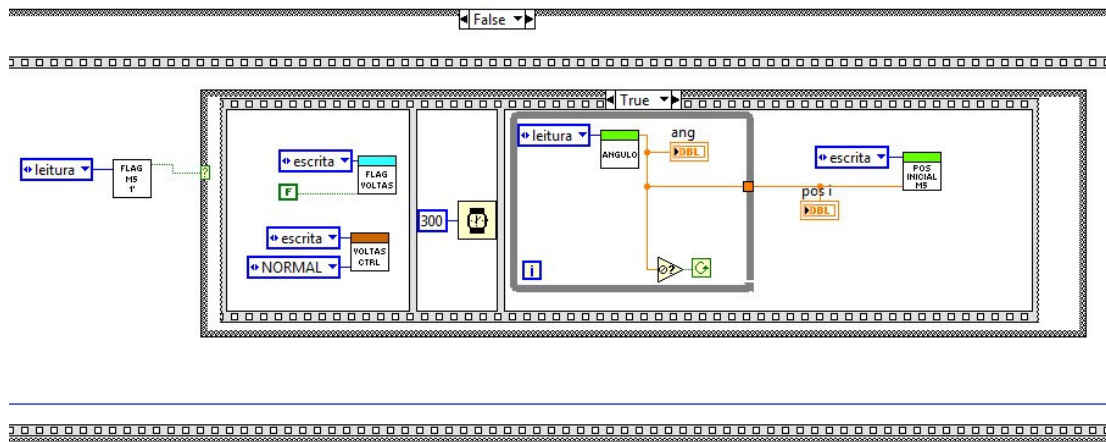


Figura 42 – Segunda tarefa no modo automático de funcionamento

Devido ao facto de na linguagem LabVIEW, sempre que uma função global é lida, é considerado que o valor lido é novo, mesmo que o mesmo se tenha mantido inalterável durante muito tempo, torna-se necessário a implementação de detectores de mudança de valores. Estes detectores são programados para lerem a função global e rejeitarem o valor lá armazenado até este ser alterado. Este detector de transição de estado, está responsável pela monitorização da flag de controlo que indica se o motor está em movimento ou se encontra parado. Caso o motor se encontre parado, pode-se proceder à activação das

restantes flags de controlo. Caso o motor esteja em movimento, não será realizada qualquer tarefa até este se imobilizar.

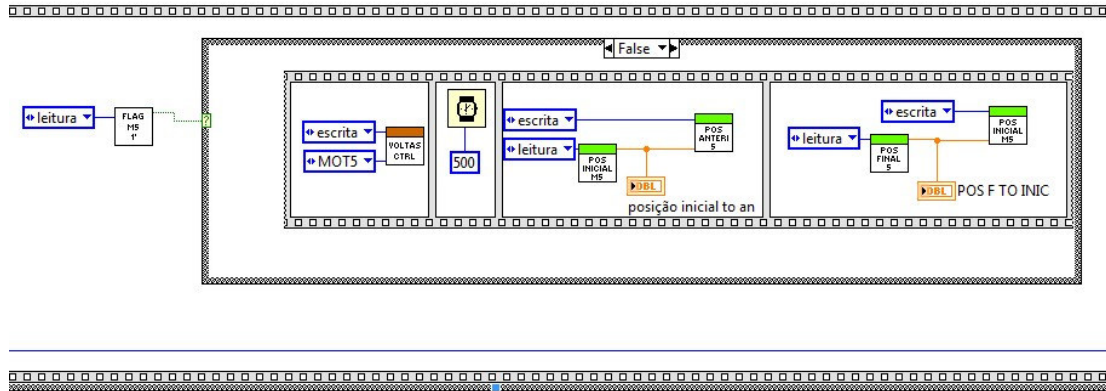


Figura 43 – Segunda tarefa após a primeira deslocação

Para memorização das posições angulares, revelou-se ser necessário distinguir duas situações distintas. A primeira situação consiste na primeira deslocação do veio do motor e a segunda situação (figura 43) consiste nas deslocações seguintes do veio do motor.

Para distinção entre as duas situações, aplicou-se uma função global que indica para cada motor se é a primeira vez que o motor irá se movimentar ou não. Caso seja a primeira vez, é necessário inicializar o número de voltas do veio do motor a zero para a correcta leitura da posição inicial. Como a função global, que armazena a posição angular actual do veio do motor, é executada a partir de dois ciclos while paralelamente, foi necessário introduzir um filtro, o qual rejeita valores não válidos para a posição angular.

Com a correcta leitura da posição inicial, activa-se o contador de voltas e coloca-se o drive no estado activo. Caso a deslocação a efectuar não seja a primeira, é inicializado o número de voltas que o mesmo motor efectuou anteriormente, a posição anterior assume o valor da posição inicial e a posição inicial assume o valor da posição final.

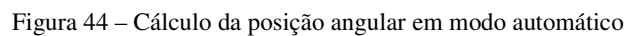


Figura 45 – Envio da posição pretendida após a primeira deslocação

Caso não seja a primeira deslocação do veio do motor (figura 45), é efectuada uma comparação. Caso a posição anterior atingida seja maior que a posição angular actual, é calculada a deslocação que o veio deve realizar para voltar à posição anterior. Esse valor é somado à posição angular actual e é enviado o resultado à rotina de controlo de velocidade do motor. Caso a posição anterior atingida seja menor que a posição angular actual, a deslocação necessária é subtraída à posição angular actual e o resultado é enviado à rotina de controlo de velocidade do motor. Esse resultado é armazenado como posição final, é activado a flag de início de movimento.

Na quarta tarefa (figura 46), está presente o controlo manual de paragem de emergência como em todos os modos de funcionamento anteriores, sendo a estrutura muito semelhante aos modos de funcionamento anteriores. Para além da comutação de sinais na ordem correcta de execução, está presente uma diferença. É nesta tarefa que é armazenado o número de voltas que o veio do motor efectuou durante o deslocamento. Este passo é a chave para se implementar uma referência absoluta, pois o contador de voltas será inicializado com outros valores assim que o controlo executa o funcionamento de outro motor.

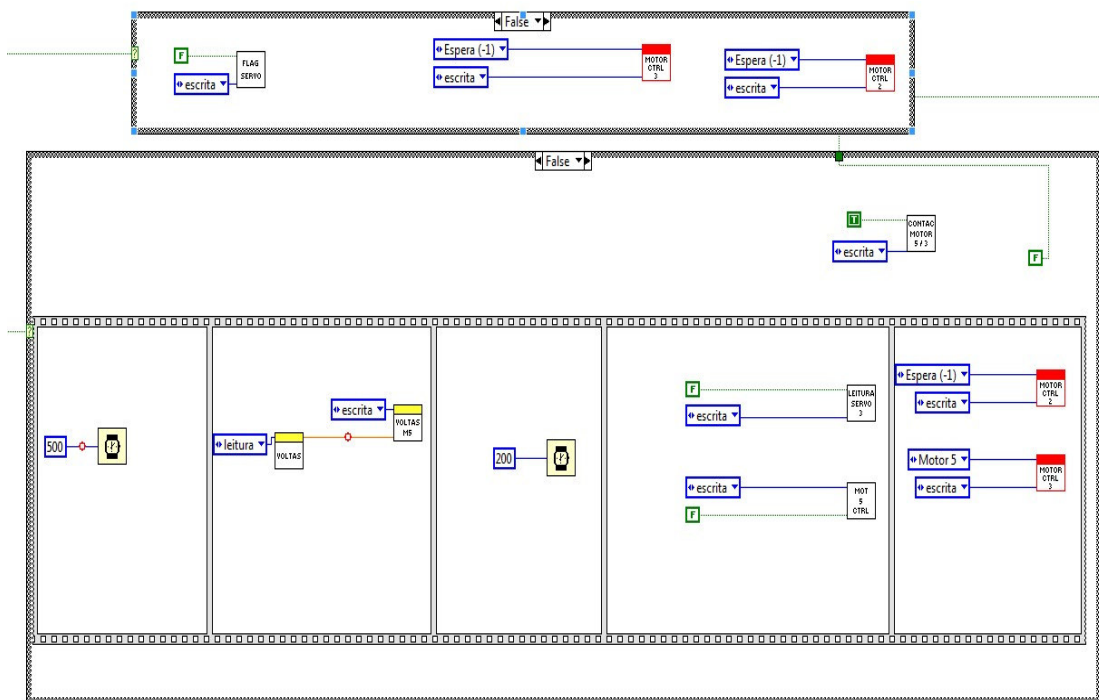


Figura 46 – Quarta fase no modo automático

7. Conclusão e trabalho futuro

Ponderando no trabalho realizado, conclui-se ser possível realizar a multiplexagem dos drives utilizando sinais analógicos. Esta multiplexagem revelou contudo ser complexa em vários aspectos.

Em primeiro lugar, verificou-se ser difícil a configuração das características eléctricas de motores diferentes no mesmo drive, sendo necessário encontrar um compromisso em termos de especificação de corrente, binário e velocidade de rotação. Desta forma, quanto mais próximas forem as características dos motores, melhor será o desempenho global do sistema.

A sincronização de todos os sinais de controlo dos drives em conjunto com a comutação de todos os sinais necessários a cada motor, também se revelou ser crítica. Foi necessário impor atrasos temporais, para que a execução do programa se mantivesse sincronizada com a propagação, comutação e actuação dos sinais eléctricos.

Uma limitação, que surgiu no decorrer desta dissertação, foi o desempenho do equipamento de controlo utilizado. Como anteriormente referido, o tempo de execução de um ciclo de computação é mais longo que o esperado, limitando a velocidade máxima que o manipulador pode atingir.

Por outro lado, quero realçar que a capacidade de manipulação de cargas manteve-se assim como a precisão dos movimentos, visto que o erro apenas variava na ordem das centésimas de milímetro.

De uma forma global, a recuperação de um manipulador industrial provou ser viável, vantajosa e mais económica. Deve-se notar também que a recuperação de um manipulador é aplicável apenas em casos específicos, dependendo do equipamento, podendo contudo não ser a melhor solução em algumas situações em que é exigido um manipulador rápido.

Como trabalho futuro, o primeiro objectivo deverá ser a aquisição de uma unidade de controlo mais rápida ou utilizar autómatos programáveis. Uma aquisição de sinais mais rápida afecta a qualidade das amostras e consequentemente o cálculo das posições angulares dos veios dos motores. Este aumento do número de amostras por segundo implicará a utilização, na unidade Crio, de outras cartas de comunicação. Outra forma de melhoramento consistiria na optimização do software, tentando diminuir o tempo de ciclo de execução, isto é aumentando o número de vezes em que a posição angular é calculada,

permitindo assim a movimentação mais rápida do manipulador. Sucedo contudo que esta operação implicaria ter acesso ao software do LabVIEW uma alteração que não se afigura fácil por desconhecermos estes programas.

Sugerimos assim a utilização de outro controlador, com mais capacidade de processamento e mais memória, como por exemplo o controlador PXI também da National Instruments cuja aquisição não se afigurou possível por razões de timing e económicas.

Bibliografia

- [1] Brushless DC motor - http://www.allaboutcircuits.com/vol_2/chpt_13/6.html
- [2] Synchronous motors - http://www.allaboutcircuits.com/vol_2/chpt_13/2.html
- [3] Voltage and frequency variation –
<http://www.usmotors.com/TechDocs/ProFacts/Voltage-Frequency-Variation.aspx>
- [4] Wikipédia - Brushless DC motor -
http://en.wikipedia.org/wiki/Brushless_DC_electric_motor
- [5] Wikipédia – DC motor - http://en.wikipedia.org/wiki/DC_motor
- [6] Resolver - http://www.admotec.com/rotasyn_resolvers.htm
- [7] Resolver - <http://amci.com/tutorials/tutorials-what-is-resolver.asp>
- [8] Resolver – Synchro and Resolver Engineering Handbook – Moog components group.
- [9] Resolver – Understanding resolvers and resolver-to-digital conversion.
- [10] Resolver – Synchro/resolver conversion handbook, DDC, fourth edition
- [11] Resolver – Resolver built-in DC Brushless Motor Control, Renesas
- [12] Bibliotecas do LabVIEW
- [13] Documentação do drive Moog